

**XML APPLICATION
IN DATABASE SERVER ACCESSING
Case Study : Computer Virus Database**

Skripsi

**Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
Jurusan Teknik Informatika**



Disusun Oleh :

Evy Sandra
NIM : 005314008



**JURUSAN TEKNIK INFORMATIKA
FALKUTAS TEKNIK
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2004**

HALAMAN PERSETUJUAN

Skripsi

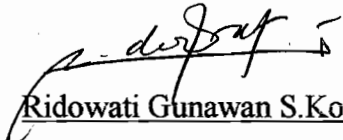
**XML APPLICATION
IN DATABASE SERVER ACCESSING
Case Study : Computer Virus Database**

Disusun oleh :

Evy Sandra
NIM : 005314008

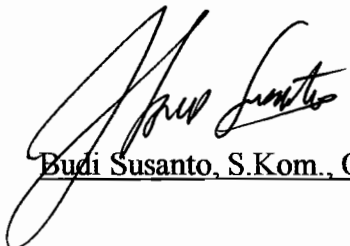
Telah disetujui oleh :

Dosen pembimbing I,


Ridowati Gunawan S.Kom., M.T.

Tanggal : 20 Juli 2004

Dosen Pembimbing II,


Budi Susanto, S.Kom., OCA

Tanggal : 19 Juli 2004

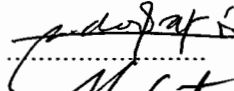
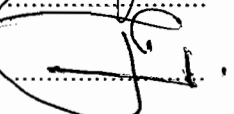
HALAMAN PENGESAHAN
SKRIPSI
XML APPLICATION
IN DATABASE SERVER ACCESSING

Dipersiapkan dan ditulis oleh :

Evy Sandra
NIM : 005314008

Telah dipertahankan di depan Panitia Penguji
Pada tanggal 8 Juli 2004
dan dinyatakan memenuhi syarat

Susunan Panitia Penguji


	Nama Lengkap	Tanda tangan
Ketua	Ridowati Gunawan, S.Kom., M.T.	
Sekretaris	Budi Susanto, S.Kom., OCA.	
Anggota	JB. Budi Darmawan, S.T., M.Sc.	
Anggota	DS. Bambang Soelistijanto, S.T.	

Yogyakarta, 21 Juli 2004

Fakultas Teknik
Universitas Sanata Dharma Yogyakarta

Dekan,




(Ir. Greg. Herliarko, SJ., SS., B.ST., MA., M.Sc.)

HALAMAN PERSEMBAHAN

Kupersembahkan karyaku ini, teristimewa untuk :

"My Boss" Jesus Chirst

Pada saat aku lemah ...

Pada saat aku hanya bisa menangis ...

Aku datang kepada - Mu

Ku temukan kelegaan dan damai yang sejati

Ya ...

Karena aku sadar aku tidak berjalan sendiri

Aku berjalan bersama SAHABAT SEJATI ku

"Kau berikan yang terbaik bagiku, maka aku ingin memberikan terbaik
bagi - Mu"

Keluarga Besar David Suwarno

Keluargaku tercinta, bersama kalian aku bisa bertahan ...

Aku bisa lebih berhikmat dan bijaksana ...

**Keluargaku, kalian yang mengingatkan aku bahwa Jesus lebih
berkuasa dari apapun juga yang ada di dunia ini.**

**"Kalian sangat berharga bagiku, Tuhan jangan pisahkan aku
dengan mereka sebelum aku bisa membahagiakan mereka"**

Antonius Sinung Wicaksono, S.T.

"Kado Natal dan Kado Ulang Tahun untukmu"

**Selamat menikmati hari - hari mu di schlumberger, aku di
sini selalu mendukungmu.**

"Aku sangat bahagia dengan segala warna cintamu"

MOTTO

Ku serahkan segala kuatirku kepada – Mu

Karena ...

Kaulah kuatku dan kebangganmu

Karena aku tahu, Bersamamu segala perkara dapat ditanggung di dalam – Mu dan tidak ada pekerjaan yang tidak dapat diselesaikan antara Kau dan aku secara bersama – sama

Karena ...

Bagi Tuhan tidak ada yang mustahil

Dan ...

Kau memberi segala sesuatu indah pada waktunya

Hingga ku bermazmur...

“Aku bersyukur kepada – Mu oleh karena kejadianmu dahsyat dan ajaib, ajaib apa yang Kau buat dan jiwaku benar – benar menyadarinya. (Mazmur 139 : 14)”

Segala Puji dan Hormat hanya bagi – Mu, My Jesus

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa yang telah memberikan hikmat, rahmat serta karunia - Nya, sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Tugas akhir ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada jurusan Teknik Informatika Fakultas Teknik Universitas Sanata Dharma.

Tugas akhir dengan judul “Aplikasi XML dalam pengaksesan *database server* dengan studi kasus *database* virus komputer ini diharapkan dapat bermanfaat bagi pihak *user* yang menginginkan untuk memperoleh data dalam *database server* ke dalam format XML sehingga dapat diakses pada aplikasi yang berbeda dan bagi pihak yang tertarik dengan aplikasi XML.

Dalam proses pengerjaan tugas akhir ini, penulis mendapatkan bantuan dan bimbingan dari berbagai pihak, maka dari itu pada kesempatan ini penulis hendak mengucapkan terima kasih dengan segala hormat kepada :

1. Bapak J.B Budi Darmawan, S.T, M.Sc, selaku ketua jurusan Teknik Informatika Universitas Sanata Dharma Yogyakarta yang telah memberikan kesempatan dan fasilitas untuk menyelesaikan studi pada Jurusan Teknik Informatika. “Iya pak, saya akan ingat kata – kata bapak untuk membantu kakak saya. Pasti pak !”
2. Ibu Rido Gunawan, S.Kom., MT, selaku dosen pembimbing I yang memberikan bimbingan, dorongan, motivasi, serta kemudahan dalam

mengerjakan tugas akhir ini. Dengan nasehat dan petuah bu rido yang memberi ketenangan dan keteduhan buat saya.

3. Bapak Budi Santoso, S.Kom., OCA, selaku dosen pembimbing II yang dengan sabar dan selalu memberikan motivasi kepada penulis dalam mengerjakan tugas akhir ini. “Gampang kok, coba aja pelajari pasti bisa...” kata – kata dari pak budi ini yang membuat saya terus mencoba dan tidak putus asa sampai terselesainya tugas akhir ini. Makasih buat pak budi yang menguatkan dengan kata – katanya, “Tenang aja vy, pasti kamu lulus!”
4. Bapak Donny Ariananda, S.T, makasih atas dukungan dan doanya. Selamat berjuang di Australia! Makasih dah banyak *share* lewat email.
5. Bapak DS. Bambang Soelistijanto, S.T, makasih pak atas segala nasehatnya. Saya ingat betul, pada hari sabtu tanggal 3 juli 2004, bapak banyak memberikan masukan dan nasehat pada saya dan saya tahu bahwa ini untuk menjadikanku terbaik dan *perfect* dalam pekerjaanku. Saya banyak belajar dari pak Antok, makasih pak.
6. Staff Laboratorium (Pak Bele, Mas Danang dan Mas Catur), terima kasih atas kerja samanya selama ini. Pak bele yang selalu bilang, mbak cvy “kapan maju pendadaran?” dan pak bele yang memberi penguatan pada saat saya *down* dan pak bele yang menemani saya untuk *setting* komputer, makasih pak. Mas danang yang pernah bilang, “kmu angkatan ’00 ngapain cepet - cepet?” dan mas catur yang pernah minjamin kartu lift. Thanks so much, God Bless U All.

7. Keluarga saya, papi, mami, mbak nie dan dek wendy. Papi yang selalu bilang “Tuhan memberkati” pada setiap kali saya mau pulang ke kost, Mami yang selalu bilang “Cepat lulus!”, Mbak nie yang selalu bilang “dek kamare diresiki” mbak nie makasih banget atas bimbingan dan kasih sayangnya buat adik kecil ini, dengan segenap hatiku aku mendoakanmu untuk cepat lulus. dek wendy yang pernah nanya “kapan kowe lulus mbak? komputere ta ngone”.
8. Antonius Sinung Wicaksono, S.T, Mas Tony yang dalam kesibukan senantiasa memberikan cintanya, kasih sayangnya, kesabarannya, nasihatnya dan bimbingannya. Dia adalah anugerah dalam hidup saya, dengan dia disampingku semakin menambah kebanggan dalam hidupku.
9. Keluarga Besar Bapak Klemens di lampung (Bapak Ibu, Mas Andre, Dek Ian), makasih atas dukungan doanya selama ini. Bapak di lampung yang selalu bilang “Doa Bapak Ibu mengiringi usahamu”.
10. Keluarga Dewi (Bapak, Ibu, Mas Bayu, Mas Wid dan si ulilku “dewi”), makasih dukungan doanya. Sungguh, aku sangat bahagia bersama kalian. Cepatan lulus ya say, aku berharap kita bisa wisuda bareng. “Wi, sampai kapanpun kamu akan menjadi sahabatku sekaligus saudaraku”
11. Pembimbing spiritualku, Mas Gianto di jakarta, makasih yang selalu mengingatkan untuk renungan pagi dan selalu berserah pada “Jesus Christ”.
12. Dewie “Ulil”, Nanda “Unchit” dan ochie “Maniez” makasih dah mau dengerin curhatan dan tangisanku.

13. Wawan “Ndundut”, Kau memang sahabatku... makasih atas persahabatannya, kita akan bertemu dalam keadaan yang lebih baik dari sekarang... kita akan menjadi orang yang sukses dan berhasil, Amien. Ambil S2 bareng yuk ndut...
14. Criss “Think No Limit”, evy bangga ma kamu...
15. Dominicus Mario Setyono, S.T “PT. Athos”, makasih atas persahabatannya selama ini.
16. Staff CV. RAS (*Rapid Application System*)... Rudy, Felix, Agus, Doddy. Rudy “Pak direktur”, makasih atas pinjaman komputernya untuk ngerjain tugas akhir.
17. Semua Pegawai “Konveksi ASTARI”, makasih atas kerjasamanya selama ini.
18. Budhe Padhe Harsono, Makasih atas doa restunya dan tempat tinggal waktu berjuang di SLTP N 2 Klaten.
19. Teman seperjuanganwaktu bimbingan (Ruby, Nicko), evy bangga ama kalian.
20. Mas Koko (’98) dan Mas Edi (’97), akhirnya kita bisa melewati dengan indah Kamis, 8 Juli 2004. Makasih buat Mas Koko yang mendukungku dengan kata – katanya “jangan menyerah sebelum berperang” dan Mas Edi dengan kata – katanya “kamu pasti lulus dan dapat nilai A”. Makasih untuk kalian semua Mas.
21. Teman baikku Wiwin, Ira, Rosa, Thesa, Yohana, Dessy, Arum, Zita, Valen, Daniel Adi, Daniel Darius, Dewa, Tikus, Irjay “Capjay”, Lius, Sean dan Arman.

Evy bahagia bersama kalian, sungguh kalian teman yang baik. Makasih ya jay buat penguatannya dengan *messagemu* “tenang aja vy, jangan gugup. Tuhan memberkati”

22. Teman – teman angkatan '00, maju terus... gapai cita – cita hingga puncak”.
23. Mas Hestu”Pak c”, yang mengajarkan evy untuk membuat keputusan dengan bijaksana.
24. Mas “Indrug”, yang telah mengajarkan evy untuk terus berjuang. Dia yang selalu tanya”kapan maju?” sehingga mendorongku cepat lulus.
25. Mas VQ, makasih pernah temenin ke UKDW ketemu pak budi.
26. Mas Agung PCJ ”Pinter Cerdas Jenius”, Mas Krismawan, Mas Wahyu, Mas Peter, Dimas “Sapi”, Mas “Jonggos”, Mas “Gogon” dan Mas Denny, makasih atas dukungan untuk menyelesaikan tugas akhir ini.
27. Teman – teman angkatan '99 (Mbak Novie, Mbak Kristin, Mbak Nuning, Mbak Elsa “Kecil”, Mbak Eko, Mbak Vivin, Mbak Arum, Cie Tika, Cie Yanti), makasih atas dukungan doanya, evy banyak belajar dari kalian.
28. Teman – teman kost Pancawati (Dek Asti, Mbak Emi, Dek ima, Dek Anjar, Dek Ranie, Dek Tere, Dek Tusti, Maria “mambut”, Dek Tina, Dek Lusi, Dek tiwie), sungguh evy bahagia bersama kalian.
29. Keluarga Besar Bapak Ibu Mudjiono “Bapak/Ibu Kost Pancawati”, Makasih doa restunya. Untuk Dek nova, makasih dah jadi adekku yang baik, dah mo dengerin curhat mbak evy. Untuk dek mufid alias “Cuprit”,

kalo ngambek jangan diem aja, kasihan ibu tuh... Untuk mbak nanie, kita pernah berjuang bareng nglamar pekerjaan, cepat married ya mbak.

Penulis menyadari bahwa laporan tugas akhir ini masih jauh dari sempurna dan banyak kekurangannya, untuk itu penulis berterimakasih atas saran dan kritikan yang membangun demi menyempurnakan tugas akhir ini. Akhirnya penulis mengharapkan semoga laporan tugas akhir ini dapat bermanfaat bagi kita semua.

Yogyakarta, 12 juli 2004



Evy Sandra

SURAT PERNYATAAN BUKTI KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa skripsi ini adalah asli hasil karya saya dan tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka, sebagaimana layaknya karya ilmiah.

Demikian surat pernyataan ini saya buat dengan sebenar – benarnya.

Yogyakarta, 12 juli 2004



Evy Sandra

ABSTRAKSI

Pada era globalisasi ini, kemudahan untuk mendapatkan informasi menjadi prioritas utama. Terlebih lagi bahwa pada saat ini *user* (pemakai) menginginkan untuk dapat mengolah sendiri terhadap data yang diperoleh. Yang menjadi masalah di sini adalah bagaimana *user* dapat memperoleh data dalam suatu *database server* ke dalam suatu format yang bersifat umum (*general*), sehingga *user* pada pihak *client* dapat mengambil dan mengolah data tersebut dalam aplikasi yang berbeda.

XML (*eXtensible Mark up Language*) menjawab kekurangan ini. Dalam hal ini akan dimanfaatkan aplikasi XML dan protokol HTTP (*Hyper Text Transfer Protocol*) dalam pertukaran data. XML akan diolah dalam mesin PHP sehingga dihasilkan dokumen XML. Dokumen XML yang dihasilkan dapat *download* pada pihak *client* dengan aplikasi yang berbeda. Dalam hal ini aplikasi *client* yang digunakan adalah Visual Basic. *Database* yang digunakan adalah MySQL. Untuk memudahkan pemodelan ini akan digunakan studi kasus *database* virus komputer.

Penggunaan dokumen XML untuk pertukaran data antar *database client* dengan aplikasi yang berbeda melalui *web server* dan PHP, diharapkan dapat memberikan suatu model pertukaran data yang bersifat umum.

ABSTRACT

In this globalization era, easy of get information access become main priority. Moreover, in this time users want information that they can process the information by themselves. The problem for user is how to obtain information from a database server in general format, so user in the client side can get and process the data in the different application.

XML (eXtensible Mark up Language) response to this lack. In this problem will be used XML application and HTTP protocol (Hyper Text Transfer Protocol) in data exchange. XML will be processed in PHP machine and produce XML documents. This XML documents can be upload at client side using different application. In this example client's application is Visual Basic. The Database used is MySQL. To simplify this modeling, computer virus database will be used as case study.

Document XML application that used for data exchange between client database using different application through web server and PHP, can give data exchange modeling generally.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTTO	v
KATA PENGANTAR	vi
PERNYATAAN KEASLIAN KARYA	xii
ABSTRAKSI	xiii
<i>ABSTRACT</i>	xiv
DAFTAR ISI	xv
DAFTAR GAMBAR	xx
DAFTAR TABEL	xxii
BAB 1 PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penulisan	6
1.5 Metodologi Penelitian	6
1.6 Sistematika Penulisan	8
BAB 2 LANDASAN TEORI	
2.1 <i>Internet</i> dan HTTP	9



2.1.1 <i>Internet</i>	9
2.1.1.1 <i>World Wide Web</i>	9
2.1.2 HTTP	10
2.2 XML	11
2.2.1 Struktur dasar XML	12
2.2.2 Sintaks XML	14
2.2.3 Jenis Dokumen XML	18
2.2.4 DTD (<i>Document Type Definition</i>) dan Validasi XML	18
2.2.5 Keuntungan dan Keunggulan XML	22
2.3 PHP	23
2.3.1 Struktur Skrip PHP	24
2.3.2 Sintaks Dalam PHP	25
2.3.2.1 Operator	25
2.3.2.2 Struktur Kontrol	27
2.3.2.3 Fungsi	31
2.3.2.4 Penangan <i>Session</i>	32
2.3.3 Konsep Kerja PHP	34
2.3.4 PHP Membuat Halaman XML	35
2.3.4.1 PHP Membuat Halaman XML Secara Dasar	35
2.3.4.2 PHP Membuat Halaman XML dari MySQL	38
2.4 MySQL	40
2.4.1 Sintaks MySQL	40
2.4.2 Tipe Data MySQL	42

2.4.3 Kelebihan MySQL	43
2.5 Visual Basic	43
2.5.1 Pengenalan Visual Basic	43
2.5.2 <i>Object Microsoft</i> DOM dalam Visual Basic	44
2.6 Informasi Virus Komputer	45
2.6.1 Definisi Virus Komputer	46
2.6.2 Karakteristik Virus Komputer	46
2.6.3 Jenis Virus Komputer	47
2.6.4 Cara Kerja Virus Komputer	48
 BAB 3 ANALISIS DAN PERANCANGAN SISTEM	
3.1 Sistem yang Dikembangkan	49
3.2 Kebutuhan Sistem	49
3.3 Analisis Sistem	50
3.3.1 Konsep Kerja Sistem	50
3.3.2 DAD (Diagram Arus Data)	53
3.3.2.1 Konteks Diagram	54
3.3.2.2 Diagram Berjenjang	55
3.3.2.3 DAD Level 0	56
3.3.2.4 DAD Level 1 (Proses 4)	57
3.3.2.5 Diagram Keseluruhan	58
3.3.2.6 Kamus Data	59
3.3.3 ERD (<i>Entity Relational Diagram</i>) Sistem	63
3.4 Perancangan Sistem	64

3.4.1 Perancangan Basis Data	64
3.4.2 Perancangan Struktur XML	69
3.4.2.1 Definisi <i>Root</i> , <i>Child</i> dan <i>Subchild</i>	69
3.4.2.2 Struktur XML	70
3.4.3 Perancangan Antar Muka	71
3.5 Analisis Sumber Daya	84
3.5.1 Sumber Daya Manusia	84
3.5.2 Perangkat Lunak (<i>Software</i>)	85
3.5.3 Perangkat Keras (<i>Hardware</i>)	85
BAB 4 IMPLEMENTASI	
4.1 Implementasi Program	88
4.1.1 Koneksi <i>Database</i>	88
4.1.2 Modul Antar Muka Pihak <i>Server</i>	89
4.1.2.1 Modul <i>Index</i> Administrator	89
4.1.2.2 Modul Utama Administrator	92
4.1.2.3 Modul <i>Add Virus</i>	95
4.1.2.4 Modul <i>Add Target</i>	96
4.1.2.5 Modul <i>Add Solution</i>	96
4.1.2.6 Modul <i>Description List</i>	97
4.1.2.7 Modul <i>Logout</i>	99
4.1.3 Modul PHP membuat halaman XML	99
4.1.4 Modul Antar Muka Pihak <i>Client</i>	104
4.1.4.1 Modul Untuk Tampilan Awal	104

4.1.4.2 Modul Menu Utama	104
4.1.4.3 Modul Setup	106
4.1.4.4 Modul <i>Information Of Virus Computer</i>	108
4.1.4.5 Modul <i>Search</i>	113
4.1.4.6 Modul <i>Help About Programmer</i>	116
4.1.4.7 Modul <i>Help About Program</i>	116

BAB 5 ANALISIS IMPLEMENTASI

5.1 Analisis Manfaat	117
5.2 Analisis Teknologi	118
5.3 Analisis Karakteristik pengguna	120
5.3.1 Analisis Karakteristik Pihak <i>Server</i>	120
5.3.2 Analisis Karakteristik Pihak <i>Client</i>	121
5.4 Kelebihan dan kekurangan program	121
5.4.1 Kelebihan Program	121
5.4.2 Kekurangan program	123

BAB 6 KESIMPULAN DAN SARAN

6.1 Kesimpulan	124
6.2 Saran	125

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar	Keterangan	Halaman
2.1	Alur Komunikasi <i>Client Server</i> Lewat HTTP Protokol	10
2.2	Struktur XML	13
2.3	Diagram Hirarki XML	15
2.4	Konsep Kerja PHP	34
3.1	Konsep Kerja Sistem	50
3.2	Konteks Diagram Informasi Virus Komputer	54
3.3	Diagram Berjenjang	55
3.4	DAD Level 0	56
3.5	DAD Level 1 (Proses 4)	57
3.6	Diagram Keseluruhan	58
3.7	ERD (<i>Entity Relational Diagram</i>) Sistem	64
3.8	Relasi Antar Tabel	65
3.9	Contoh Data dalam Relasi Antar Tabel	68
3.10	Struktur XML dalam bentuk <i>tree</i>	70
3.11	Rancangan antar muka menu <i>login</i> administrator	72
3.12	Rancangan antar muka <i>form</i> utama administrator	73
3.13	Rancangan antar muka <i>form</i> tambah data virus komputer	74
3.14	Rancangan antar muka tambah data sasaran virus komputer	75
3.15	Rancangan antar muka tambah data solusi	76

3.16	Rancangan antar muka <i>description list</i>	77
3.17	Rancangan antar muka <i>logout</i> program	78
3.18	Rancangan halaman XML pada <i>internet explorer</i>	78
3.19	Rancangan tampilan awal	79
3.20	Rancangan antar muka meriu utama	79
3.21	Rancangan antar muka untuk <i>setup</i>	80
3.22	Rancangan antar muka untuk informasi virus komputer	81
3.23	Rancangan antar muka untuk <i>form</i> pencarian data virus komputer berdasarkan nama virus	81
3.24	Rancangan antar muka untuk <i>form</i> pencarian data virus komputer berdasarkan bulan aktif	82
3.25	Rancangan antar muka untuk <i>form</i> pencarian data virus komputer target serang virus	83
3.26	Rancangan antar muka <i>form about programmer</i>	83
3.27	Rancangan antar muka <i>form about program</i>	84

DAFTAR TABEL

Tabel	Keterangan	Halaman
2.1	<i>Entity Reference</i>	12
2.2	Operator Aritmatika	26
2.3	Operator Logika	26
2.4	Operator Perbandingan	27
2.5	Tipe – Tipe data dalam MySQL	42
3.1	Struktur tabel virus_komputer	66
3.2	Struktur tabel serang	66
3.3	Struktur tabel sasaran_serang	66
3.4	Struktur tabel memiliki_solusi	66
3.5	Struktur tabel solusi	67
3.6	Struktur tabel admin	67
5.1	Manfaat pembuatan sistem aplikasi XML	117

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada era globalisasi ini, kemudahan untuk mendapatkan informasi menjadi prioritas utama. Terlebih lagi bahwa saat ini *user* (pemakai) menginginkan untuk dapat mengolah sendiri terhadap data yang diperoleh. Maka dari itu data harus dikelola secara efisien, efektif dan maksimal sehingga *user* dapat mengakses data secara mudah dan cepat. Dalam hal ini, keterlibatan komputer sangat penting sebagai pengelolaan data agar data dapat diakses, didistribusikan dan digunakan secara mudah dan cepat.

Data dapat ditampung dalam *database client* dan *database server*. *Database client* merupakan *database* yang diciptakan untuk *single user*. *Database server* merupakan *database* yang diciptakan untuk *multi user*. Suatu data yang disimpan dalam *database server* akan diakses oleh pihak *client* dengan menggunakan koneksi *database*. Secara umum koneksi *database* yang baik adalah dapat mengkoneksikan *database* ke pihak *client* secara mudah dan cepat. Pada saat ini, banyak *user* yang menginginkan untuk memperoleh data dari *database server* dalam format umum (*general*) sehingga dapat diakses pada aplikasi yang berbeda. Jadi intinya, bagaimana memperoleh data dalam suatu format umum sehingga *user* dapat mengambil dan mengolah data dalam aplikasi yang berbeda.

XML menjawab kekurangan ini. XML (*eXtensible Mark up Language*) adalah *mark up language* yang menyediakan format untuk mendeskripsikan data

terstruktur atau terurut. XML menyediakan fasilitas yang membuat isi (*content*) sebuah data lebih mudah dimengerti. Dalam XML, apa yang diinginkan oleh pemberi informasi, yaitu pembuat *file* XML akan dibaca penerima informasi dengan jelas. Dengan menggunakan XML akan disajikan informasi dalam dokumen XML yang dapat diakses dalam aplikasi yang berbeda.

Pada penulisan tugas akhir ini akan mengambil topik “Aplikasi XML Dalam Pengaksesan *Database Server* dengan Studi Kasus *Database Virus Komputer*“. Sistem yang akan dibuat adalah sistem mengenai aplikasi XML dalam pengaksesan *database server* yang mengubah data dalam suatu *database* ke dalam format dokumen XML, sehingga data dapat diakses pada aplikasi berbeda dengan memanfaatkan protokol HTTP (*Hyper Text Transfer Protocol*) sebagai tempat untuk pertukaran data. Setelah selesai dibuatnya program aplikasi XML ini diharapkan nantinya dapat memberikan suatu model pertukaran data yang bersifat umum.

1.2 Rumusan Masalah

Inti permasalahan dalam penulisan tugas akhir ini adalah sebagai berikut :

“Bagaimana caranya *user* dapat memperoleh data dari *database server* ke dalam suatu format dokumen XML, sehingga *user* dapat mengambil dan mengolah data pada aplikasi yang berbeda dengan memanfaatkan protokol HTTP dalam melakukan pertukaran data ?“

1.3 Batasan Masalah

Dalam penulisan tugas akhir ini, batasan masalah meliputi hal – hal sebagai berikut :

1. Sistem yang dibuat menitikberatkan pada aspek pertukaran data dari *database server* ke dalam format dokumen XML sehingga *user* dapat mengambil dan mengolah data pada aplikasi yang berbeda.
2. Perangkat lunak yang digunakan ada dua bagian, yaitu perangkat lunak di *server* dan perangkat lunak di *client*. Perangkat lunak di *server* menggunakan PHP yang menghasilkan dokumen XML dan perangkat lunak di *client* menggunakan Visual Basic 6.0 yang digunakan sebagai antarmuka (*interface*) dalam menampilkan informasi.
3. Perangkat lunak PHP yang menghasilkan dokumen XML dapat langsung dipanggil di *browser* dalam tampilan format dokumen XML.
4. Aplikasi XML yang dibuat merupakan jenis dokumen XML validasi dengan menggunakan DTD (*Document Type Definition*) sebagai pendeklarasian data dalam dokumen XML.
5. Aplikasi yang dibuat tidak membahas masalah *security* atau keamanan dalam pertukaran data.
6. Sistem dibatasi pada aspek pengaksesan *database server* ke dalam format dokumen XML dengan studi kasus *database virus komputer*, *input data virus komputer* sangat terbatas karena keterbatasan *resources* terhadap data virus komputer.

7. Fasilitas yang disediakan pada bagian *server* untuk administrator adalah sebagai berikut :

a. *Add Virus.*

Fasilitas yang digunakan untuk menambah virus komputer.

b. *Add Target.*

Fasilitas yang digunakan untuk menambah target atau sasaran serang virus komputer.

c. *Add Solution.*

Fasilitas yang digunakan untuk menambah solusi virus komputer.

d. Fasilitas yang digunakan untuk melihat semua data tentang virus komputer, target virus komputer dan solusi virus komputer yang telah dimasukkan oleh administrator.

e. Fasilitas yang digunakan untuk mengubah data virus komputer, target virus komputer dan solusi virus komputer.

f. Fasilitas yang digunakan untuk menghapus data virus komputer, target virus komputer dan solusi virus komputer.

8. Fasilitas yang disediakan pada bagian *client* untuk *user* adalah sebagai berikut :

a. Tampilan XML dalam bentuk *treeview*.

Fasilitas tampilan data XML dalam *treeview* sehingga *user* pada *client* dapat melihat bentuk *root*, *child* dan *subchild* dari XML aslinya.

- b. Tampilan XML dalam bentuk *text box*.

Fasilitas tampilan data XML dalam *text box* sehingga *user* pada *client* dapat secara mudah melihat data – data virus komputer.

- c. Daftar semua data - data virus komputer.

Fasilitas yang memuat semua data – data virus komputer dalam bentuk *MSFlexGrid* sehingga *user* pada *client* dapat melihat semua daftar mengenai data – data virus komputer.

- d. Fasilitas *searching* berdasarkan *virus_name*, *active_time* dan target virus komputer.

- e. *Show All*.

Fasilitas yang digunakan untuk menampilkan semua data – data virus komputer secara lengkap pada model *treeview*. Pada saat *user* melakukan *search* maka *user* akan mendapat data virus komputer yang dikehendaki dan untuk mengembalikan data secara lengkap seperti semula maka *user* menekan tombol *show all*.

9. Pada bagian *client*, yaitu pada aplikasi Visual Basic hanya digunakan untuk menampilkan informasi virus komputer dengan fasilitas *search* berdasarkan *virus_name*, *active_time* dan target virus komputer. Jadi intinya pada bagian *client* bukan sistem informasi atau sistem pakar yang secara detail dapat mensolusikan komputer yang terserang virus komputer.
10. Pada bagian *client*, data virus komputer ditujukan bagi *user* yang telah mengenal virus komputer.

1.4 Tujuan Penulisan

Maksud dan tujuan dari penulisan tugas akhir “Aplikasi XML Dalam Pengaksesan *Database Server* dengan Studi Kasus *Database Virus Komputer*“ ini adalah sebagai berikut :

1. Menerapkan penggunaan aplikasi XML atau dokumen XML untuk pengaksesan *database server* sehingga diperoleh data dalam format dokumen XML dan dapat diakses pada aplikasi yang berbeda dengan memanfaatkan protokol HTTP dalam melakukan pertukaran data.
2. Memberikan panduan secara teori dalam pengaksesan *database server* ke dalam format dokumen XML sehingga dapat diakses pada aplikasi yang berbeda.
3. Sebagai salah satu syarat administratif, untuk memperoleh gelar kesarjanaan S1 di Fakultas Teknik Jurusan Teknik Informatika Universitas Sanata Dharma.

1.5 Metodologi Penelitian

Metodologi yang akan digunakan dalam penyusunan tugas akhir ini adalah sebagai berikut :

1. Peninjauan (*Survey*) kebutuhan program.
Mengumpulkan data – data mengenai virus komputer dan mencari bahan - bahan yang mendukung program yang akan dibuat.
2. Studi Literatur, meliputi :
 - a. Pemahaman dan pendalaman konsep.

Yaitu memahami dan mendalami konsep – konsep aplikasi XML dalam pengaksesan *database server*, sehingga diperoleh data dalam format dokumen XML yang dapat diakses pada aplikasi berbeda dengan memanfaatkan protokol HTTP dalam melakukan pertukaran data.

b. Mempelajari perangkat lunak yang terlibat.

Pembelajaran ini meliputi pemahaman dalam pengaksesan *database server*, yaitu My SQL untuk dapat diubah ke dalam format dokumen XML dengan memanfaatkan PHP yang menghasilkan dokumen XML dan pemahaman mengenai *load* dokumen XML ke dalam aplikasi Visual Basic.

3. Analisis sistem yang akan dikembangkan.

Menganalisis sistem yang akan dikembangkan, kebutuhan sistem, konsep kerja sistem, DAD (Diagram Arus Data) dan ERD (*Entity Relational System*).

4. Perancangan sistem.

Memulai merancang desain *database*, struktur XML dan desain perancangan antar muka program yang akan dibuat.

5. Pembuatan program.

Mengkonversikan semua rancangan berdasarkan rancangan desain yang telah dibuat.

1.6 Sistematika Penulisan

Sistematika yang dipakai dalam penulisan tugas akhir ini terbagi atas enam bagian dengan garis besarnya sebagai berikut :

BAB 1 PENDAHULUAN

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penulisan, metodologi penelitian dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini berisi landasan teori yang dipakai untuk pembahasan penulisan tugas akhir.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab ini merupakan bab yang membahas tentang analisis dan perancangan sistem yang akan dibuat.

BAB 4 IMPLEMENTASI SISTEM

Bab ini berisi implementasi tiap tampilan atau antarmuka dari rancangan yang dibuat pada bab sebelumnya dan menganalisis program pada masing – masing tampilan atau antarmuka.

BAB 5 ANALISIS IMPLEMENTASI

Bab ini berisi tentang analisis implementasi dari aplikasi yang dibuat dan berisi tentang kelebihan dan kekurangan program.

BAB 6 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang menjawab tujuan tugas akhir dan berisi saran yang berguna untuk pengembangan program.

BAB 2

LANDASAN TEORI

Secara garis besar yang akan diuraikan dalam landasan teori adalah pengenalan mengenai *internet* dan HTTP (*Hyper Text Transfer protocol*), XML (*eXtensible Mark up Language*), PHP (*Personal Home Page tools*), My SQL, Visual Basic dan informasi mengenai virus komputer.

2.1 *Internet* dan HTTP

2.1.1 *Internet*

Internet atau *Interconnecting Networking* adalah dua komputer atau lebih yang saling berhubungan membentuk jaringan komputer hingga meliputi jutaan komputer di dunia (*international*), yang saling berinteraksi dan bertukar informasi. Jaringan – jaringan tersebut saling berhubungan atau berkomunikasi satu sama lain dengan berbasiskan protokol IP (*Internet Protokol*) dan TCP (*Transmission Control Protocol*), sehingga setiap *user* dari setiap jaringan dapat saling mengakses semua layanan yang disediakan oleh jaringan lainnya.

2.1.1.1 *World Wide Web*

World Wide Web merupakan sebuah arsitektur untuk memasuki dokumen – dokumen yang saling berhubungan di seluruh *internet*. Untuk saat ini, *World Wide Web* merupakan layanan yang paling sering digunakan dan memiliki perkembangan yang sangat cepat karena dengan layanan ini *user* bisa menerima informasi dalam berbagai format (*multimedia*). Untuk megakses layanan *World*

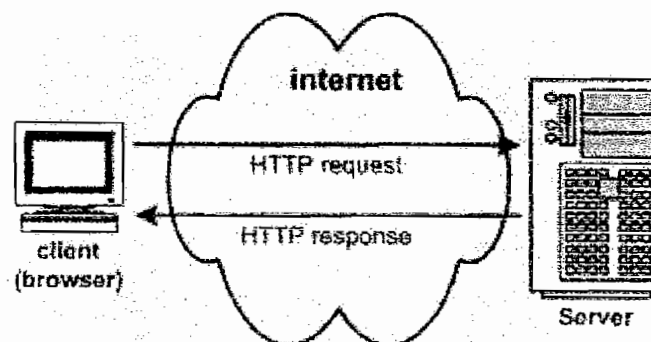
Wide Web dari sebuah komputer (yang disebut *WWW server* atau *web server*) digunakan program *web client* yang disebut *web browser* atau *browser* saja. Jenis-jenis *browser* yang sering digunakan adalah: *Netscape Navigator/Communicator*, *Internet Explorer*, *NCSA Mosaic*, *Arena*, *Lynx*, dan lain-lain.

2.1.2. HTTP (*Hyper Text Transfer Protocol*)

HTTP (*Hyper Text Transfer Protocol*) adalah protokol pada lapisan aplikasi yang dipergunakan dalam WWW (*World Wide Web*). Dengan protokol ini sebuah *web client (browser)* dapat melakukan pertukaran data dengan *web server*.

Web server bertugas melayani permintaan *browser* terhadap halaman *web* dan dokumen lain. *Web server* biasa dijalankan sebagai suatu *server* yang siap melayani permintaan setiap akan dibutuhkan.

Proses permintaan (*request*) dan pengiriman (*response*) dokumen *website* dalam dunia *internet* menggunakan protokol HTTP. Alur komunikasi antara *browser* dengan *web server* tampak seperti Gambar 2.1.



Gambar 2.1 Alur Komunikasi *Client-Server* Lewat *HTTP Protocol*
(Gregorius Agung, *Web dalam Layar Ponsel*, Yogyakarta, C.V. Jubilee Solusi Enterprise, 2001, hlm. 39)

Pada alur Gambar 2.1 dapat dijelaskan sebagai berikut :

1. Proses HTTP *Request*.

Client (browser) meminta informasi kepada *server*.

2. Proses HTTP *Response*.

Setelah *request* tersebut diterima oleh *server* maka *server* akan mencari dokumen – dokumen HTML atau XML sesuai dengan permintaan. Jika dokumen atau informasi yang diinginkan ada di *web server*, maka *server* akan mengirim informasi tadi ke pihak *client*.

2.2 XML

XML (*eXtensible Markup Language*) adalah *mark up language* yang menyediakan format untuk mendeskripsikan data terstruktur atau terurut. Fasilitas yang disediakan XML ini membuat isi (*content*) sebuah data menjadi lebih mudah dimengerti. XML dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C (*World Wide Web Consortium*) pada bulan Februari 1998. Teknologi yang digunakan pada XML sebenarnya bukan teknologi baru, tapi merupakan *subset* dari SGML (*Standard Generalized Mark up Language*) yang telah dikembangkan pada awal 80 - an dan telah banyak digunakan pada dokumentasi teknis proyek - proyek berskala besar.

Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan *tag* pembuka (diawali dengan '<' dan diakhiri dengan '>'), *tag* penutup (diawali dengan '</' diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam *tag* pembuka). Hanya bedanya, HTML mendefinisikan dari awal *tag* dan

atribut yang dipakai didalamnya, sedangkan pada XML *user* bisa menggunakan *tag* dan atribut sesuai kehendak *user*.

Tag – tag dalam XML *case sensitive*, sehingga *tag* <Buku> berbeda dengan *tag* <buku>. Jika dalam penulisan program di dalam XML sama dengan beberapa tanda yang dipakai di XML, maka penulisan program akan memanfaatkan *entity reference* untuk mengkodekan beberapa karakter. Beberapa *entity reference* akan ditunjukkan pada Tabel 2.1.

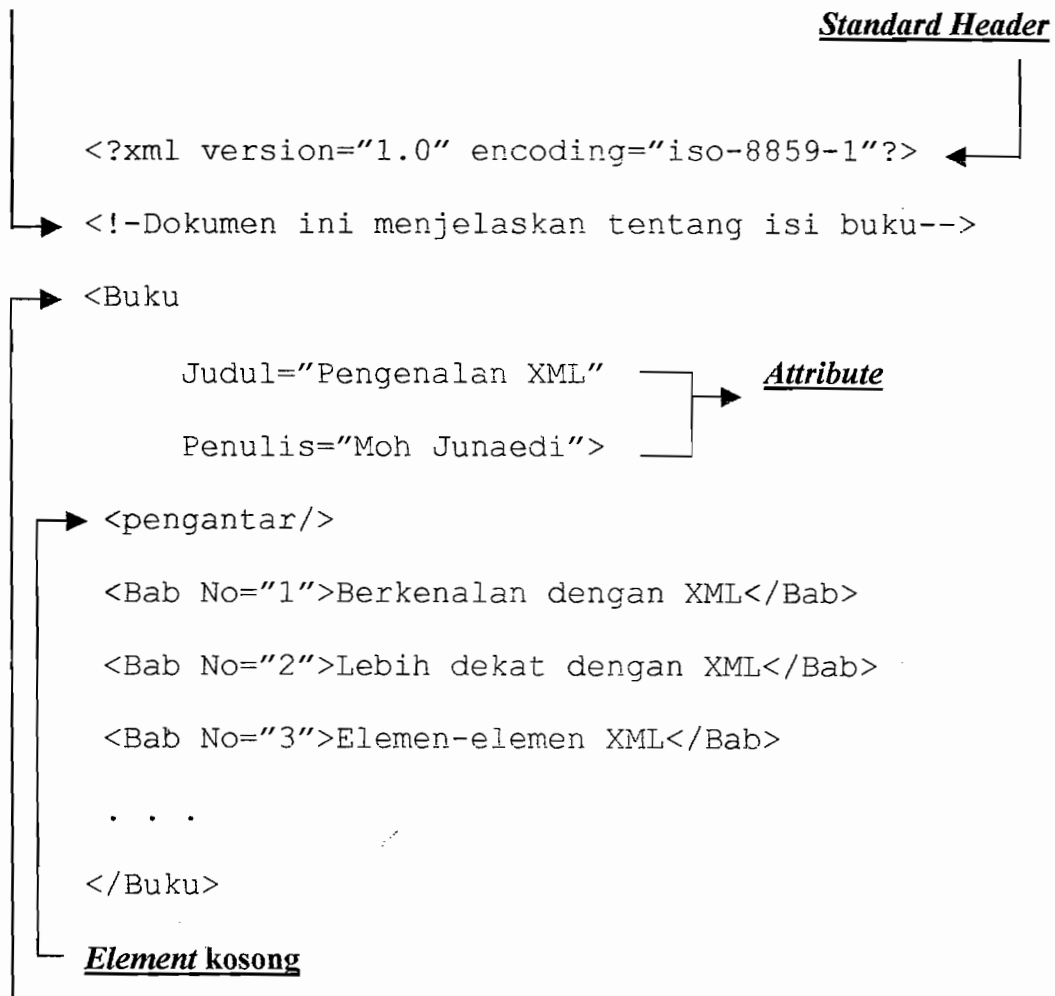
Tabel 2.1 *Entity Reference*

<i>Entity Reference</i>	Karakter
>	>
<	<
&	&
"	“
'	’

XML *parser* adalah sebuah program untuk menguji apakah sebuah dokumen XML benar atau tidak, sesuai dengan aturan XML atau tidak.

2.2.1 Struktur Dasar XML

Struktur dasar dokumen XML pada Gambar 2.2 berikut ini.

Comment**Rootnode**

Gambar 2.2 Struktur XML
(Moh Junaedi , Pengantar XML , IlmuKomputer.Com, hlm 7)

Sebuah struktur atau dokumen XML terdiri dari bagian yang disebut dengan *node*. *Node - node* itu adalah:

1. **Root node** adalah *node* yang melingkupi keseluruhan dokumen. Dalam satu dokumen XML hanya ada satu *root node*. *Node - node* yang lainnya berada di dalam *root node*.

2. **Element node** adalah bagian dari dokumen XML yang ditandai dengan *tag* pembuka dan *tag* penutup, atau bisa juga sebuah *tag* tunggal elemen kosong seperti `<anggota nama="evy"/>`. *Root node* biasa juga disebut *root element*.
3. **Attribute node** adalah *node* yang termasuk nama dan nilai atribut ditulis pada *tag* awal sebuah elemen atau pada *tag* tunggal.
4. **Text node** adalah *text* yang merupakan isi dari sebuah elemen, ditulis diantara *tag* pembuka dan *tag* penutup.
5. **Comment node** adalah baris yang tidak dieksekusi oleh *parser*.

2.2.2. Sintaks XML

Dibandingkan dengan HTML, XML lebih banyak aturan. Kalau *user* menulis sebuah dokumen HTML, beberapa kesalahan penulisan masih ditolerir. Misalnya *user* menempatkan *tag* bersilangan seperti `<p>Huruf Tebal</p>` meskipun tidak dianjurkan, HTML masih bisa bekerja dan menampilkan hasil seperti yang *user* inginkan. Tidak demikian dengan XML. XML harus memperhatikan sintaks – sintaks yang tidak boleh dilanggar, sintaks – sintaks tersebut antara lain :

1. *Heading standard* untuk *Document XML*.

Setiap membuat dokumen XML diawali dengan *heading standard XML*.

Formatnya adalah sebagai berikut:

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

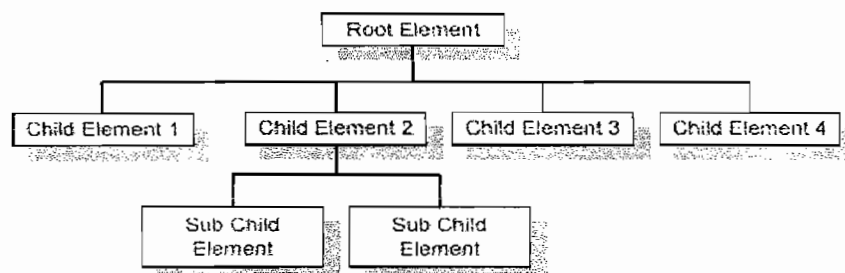
Penjelasan : `xml version="1.0"` berarti bahwa *file* XML ini dibuat sesuai dengan aturan versi 1.0 dan *encoding* menunjukkan bahwa *file* XML ini ditulis dengan kode iso-8859-1.

2. Dokumen XML harus memiliki *Root tag*.

Sebuah dokumen XML yang baik harus memiliki *root tag*. Yaitu *tag* yang melingkupi keseluruhan dari dokumen. *Tag - tag* yang lain, disebut *child tag*, berada didalam *root* membentuk hirarki. Formatnya adalah sebagai berikut :

```
<root>
  <child>
    <subchild></subchild>
  </child>
</root>
```

Bentuk mengenai diagram hirarki XML tampak pada Gambar 2.3.



Gambar 2.3 Diagram Hirarki XML

(Moh Junaedi, Pengantar XML, IlmuKomputer.Com, hlm 8)

3. *Tag* pada XML harus lengkap berpasangan.

Pada HTML beberapa elemen tidak harus berpasangan. Contoh berikut ini diperbolehkan dalam penulisan HTML.

<p>paragraph pertama

<p>paragarap kedua

Dalam XML, beberapa elemen harus berpasangan. *User* harus menulis *tag* pembuka dan penutup untuk setiap *tag* yang dibuat. Format penulisan elemen berpasangan adalah sebagai berikut :

<p>paragraph pertama</p>

<p>paragarap kedua</p>

Tag tunggal hanya diperbolehkan untuk elemen kosong. Contoh penulisan elemen kosong adalah sebagai berikut :

<anggota nama="evy"/>

4. XML membedakan huruf besar dengan huruf kecil.

Pada XML, <tanggal> berbeda dengan <Tanggal>. *Tag* pembuka dan *tag* penutup harus sama susunan huruf besar dan kecilnya. Format penulisannya adalah sebagai berikut :

<contoh>ini penulisan yang salah</Contoh>

<contoh>ini baru betul</contoh>

5. Penyarangan *tag* harus benar.

Penulisan *tag* pada XML harus mengikuti aturan *Last In First Out (LIFO)*. Pada XML *user* tidak bisa membuat *tag* yang saling bersilang seperti format berikut :

<p>Huruf Tebal</p>

tapi harus disusun seperti format berikut ini :

<p>Huruf Tebal</p>

bila dipaksakan juga, *browser* akan menampilkan pesan *error*.

6. XML mempertahankan spasi seperti apa adanya.

Berbeda dengan HTML, XML menampilkan spasi persis seperti data yang ditulis. Misalnya dalam HTML terdapat kalimat sebagai berikut :

Kami pergi bersama

akan ditampilkan sebagai berikut ini :

Kami pergi bersama

Dalam contoh kalimat 'Kami pergi bersama' meskipun digunakan spasi beberapa kali, tetapi akan ditampilkan sekali saja. Sedangkan pada XML akan ditampilkan sama persis dengan kalimat asalnya.

7. Nilai atribut harus diletakkan diantara tanda petik.

Seperti HTML, XML memiliki atribut. Nilai atribut harus diletakkan diantara dua tanda petik. Tanda petik yang digunakan bisa tanda petik tunggal ataupun tanda petik ganda. Contoh format penulisannya sebagai berikut :

`<pesan dari="evy">` atau

`<pesan dari='evy'>`

8. Penamaan *tag* dan atribut.

Nama *tag* bisa terdiri dari huruf, angka dan *underscore* ("_"). Karakter awal nama *tag* harus berupa huruf atau *underscore* ("_"), tidak diawali dengan kata xml atau XML, (misal:`<xmlstring>`), dan tidak mengandung spasi. Aturan penamaan atribut sama dengan aturan penamaan *tag*.

9. Menyisipkan komentar.

Pada bahasa pemrograman atau *scripting* kita dikenal adanya komentar (*comment*). Komentar adalah kalimat antara baris yang tidak dieksekusi oleh *compiler*, *browser* atau *parser*. Untuk menyisipkan komentar pada dokumen XML caranya adalah sebagai berikut:

```
<!--Baris ini tidak di eksekusi oleh parser -->
```

2.2.3 Jenis Dokumen XML

Dokumen XML memiliki tiga jenis dokumen XML, yaitu :

1. *Invalid Document* adalah suatu dokumen XML yang tidak mengikuti aturan penulisan yang didefinisikan oleh spesifikasi XML.
2. *Valid Document* adalah suatu dokumen XML yang mengikuti aturan spesifikasi XML dan aturan struktur dokumen yang tertuang pada DTD atau *Schema*.
3. *Well - Formed Document* adalah suatu dokumen XML yang mengikuti aturan spesifikasi XML tetapi tidak memiliki DTD ataupun *Schema*.

2.2.4 DTD (*Document Type Definition*) dan Validasi XML

Dokumen XML yang *valid* adalah yang memenuhi aturan DTD. DTD adalah suatu aturan yang mendefinisikan bagaimana seharusnya struktur isi suatu dokumen XML. Sesuai namanya DTD berfungsi untuk mendefinisikan tipe dokumen XML. Pada saat mempelajari salah satu bahasa pemrograman atau *scripting*, *user* diperkenalkan dengan deklarasi variabel, deklarasi fungsi dan deklarasi tipe data. Serupa dengan itu, DTD mendefinisikan struktur dokumen XML dengan daftar elemen yang digunakan.

Unsur - unsur yang dideklarasikan dalam DTD adalah semua unsur yang membentuk suatu dokumen XML, yaitu :

1. **Element**, satu blok data yang diawali *tag* pembuka dan *tag* penutup. Bila sebuah *element* mengandung beberapa *child element*, maka *user* perlu mendeklarasikan *child element* apa saja yang dipunyai *element* tersebut. Contoh penulisan sintaksnya adalah sebagai berikut :

```
<!ELEMENT organisasi (anggota)>
<!ELEMENT anggota (nama,alamat,kelamin,jabatan)>
```

Dalam penulisan *element* biasanya digunakan tanda – tanda yang memiliki maksud yang biasa disebut *cardinality*. *Cardinality* yang biasanya digunakan dalam penulisan *element* adalah sebagai berikut :

1. [none] artinya *default* (hanya 1)
2. ? artinya 0..1
3. * artinya 0,1..N
4. + artinya 1..N

Contoh penulisan *element* dengan menggunakan *cardinality* adalah sebagai berikut :

```
<!ELEMENT organisasi (anggota)+>
```

artinya bahwa *node* anggota bisa memiliki anggota 1..N.

2. **Attribute**, informasi pendukung *element* yang disertakan pada *tag* pembuka. Agar dokumen XML *valid*, *user* juga perlu mendefinisikan semua *attribute* yang akan digunakan dalam dokumen. Untuk mendefinisikannya digunakan *Attribute list declaration*. Contoh penulisan *attribute* adalah sebagai berikut :

```
<!ATTLIST namaelemen attribute tipe pengubah >
```

namaelemen adalah nama *element* dimana *attribute* itu digunakan. Sedangkan tipe dan pengubah adalah serangkaian informasi tentang *attribute* itu. Tipe *attribute* yang biasanya digunakan adalah sebagai berikut :

1. CDATA.

Biasanya digunakan untuk pendefinisian karakter.

2. *Enumeration*.

Sintaks enumeration adalah (nilai 1| nilai 2| nilai 3).

3. ID, IDREF, NMTOKEN, NMTOKENS, ENTITY, ENTITIES, NOTATION.

Pengubah yang biasanya digunakan dalam penulisan *attribute* adalah sebagai berikut :

1. #IMPLIED artinya *attribute* bisa ada atau tidak ada.
2. #REQUIRED artinya *attribute* harus ada.
3. #FIXED "Nilai" artinya *attribute* ada dan harus memiliki nilai tersebut.
4. Nilai default dari nilai *enumeration*.

3. **Entity**, karakter pengganti untuk sekumpulan informasi yang didefinisikan.

Dengan menggunakan *entity XML user* bisa menggantikan kalimat yang panjang atau satu blok elemen yang sering digunakan dengan sebuah pengenal singkat. Sekali *entity* didefinisikan di dalam DTD, *user* bisa menggunakannya dimana saja pada seluruh dokumen XML. Contoh penulisan *entity* adalah sebagai berikut :

```
<!ENTITY judul "Manajemen data dan informasi dengan XML/XSLT">
```

Penulisan DTD dapat dilakukan dengan dua cara, yaitu :

1. *Internal DTD*

Internal DTD merupakan penulisan DTD dengan cara menuliskan sintaks DTD dalam satu dokumen dengan dokumen XML. Penulisan DTD di dalam dokumen XML dapat dilihat pada contoh berikut :

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to      (#PCDATA)>
  <!ELEMENT from    (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body    (#PCDATA)>
]>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend</body>
</note>
```

Penjelasan dari struktur dokumen *internal DTD* dalam dokumen XML tersebut adalah sebagai berikut :

1. DOCTYPE merupakan elemen puncak. DOCTYPE dalam program tersebut adalah *note*. Suatu dokumen XML hanya boleh memiliki tepat satu DOCTYPE.
2. Tag <note> memuat empat elemen, yaitu *to*, *from*, *heading* dan *body*.
3. Elemen *to* berisi data.
4. Elemen *from* berisi data.
5. Elemen *heading* berisi data.
6. Elemen *body* berisi data.

PCDATA singkatan dari *Parsed Character Data* adalah isi data.

2. *External DTD*.

DTD dapat pula diletakkan pada *file* terpisah dari dokumen XML. Ini berguna bila DTD tersebut akan dipergunakan atau diacu oleh sejumlah

dokumen XML. Contoh penulisan *external* DTD adalah sebagai berikut

```
<!ELEMENT note (to,from,heading,body) >
<!ELEMENT to (#PCDATA) >
<!ELEMENT from (#PCDATA) >
<!ELEMENT heading (#PCDATA) >
<!ELEMENT body (#PCDATA) >
```

Penjelasan mengenai penulisan *external* DTD sama dengan penjelasan pada penulisan *internal* DTD. *File external* DTD akan disimpan dalam *note.dtd* dan akan dipanggil dalam dokumen XML. Pemanggilan DTD dalam dokumen XML dapat dilihat sebagai berikut :

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Sintaks yang digunakan untuk memanggil *file external* DTD adalah sebagai berikut :

```
<!DOCTYPE note SYSTEM "note.dtd">
```

2.2.5 Keuntungan dan Keunggulan XML

Berikut adalah keuntungan dan keunggulan XML :

1. *Ekstensibilitas*.

Artinya bebas menentukan *tag – tag* sendiri sesuai dengan kebutuhan *user*.

2. Memisahkan data dengan presentasi.

Ini merupakan keunggulan lain dari XML, yaitu pemisahan antara data dan presentasi. Maksudnya, sebuah data itu hanya berisi data saja, tidak ada

informasi lain mengenai cara menampilkannya. Sebuah dokumen XML merupakan data yang menerangkan data itu sendiri. *Tag – tag* pada XML menjelaskan mengenai isi datanya. HTML sendiri mencampuradukkan antara data dan presentasi. XML hanya mendefinisikan tentang isi saja. Misalnya ada *tag* di HTML yang menjelaskan bagaimana *browser* menampilkan data untuk ditulis miring atau tebal.

Contoh :

```
<I>Bandung</I>
```

Namun tag di XML menjelaskan data itu sendiri.

```
<nama>Bandung</nama>
```

Di contoh HTML di atas, tidak diketahui apakah Bandung kota atau julukan ataupun nama. Namun di XML, jelas bahwa Bandung itu nama.

3. Penyederhanaan Aplikasi.

Saat ini, banyak aplikasi memiliki ukuran besar dan menghabiskan memori. Biasanya aplikasi untuk mengolah kata maupun gambar dan harus mengakomodasikan berbagai format *file*. XML yang terbuka, akan memudahkan aplikasi untuk mengolah data.

2.3 PHP

Menurut dokumen resmi PHP, PHP singkatan dari PHP *Hypertext Preprosesor*. PHP itu sendiri singkatan dari *Personal Home Pages tools*. PHP merupakan bahasa berbentuk skrip yang ditempatkan di dalam *server* dan diproses di *server*. Hasilnya akan dikirimkan ke *client*, tempat *user* menggunakan *browser*.

Secara khusus, PHP dirancang untuk membentuk *web* dinamis. Artinya, dapat membentuk suatu tampilan berdasarkan permintaan terkini. Pada awalnya, PHP dirancang untuk diintegrasikan dengan *web server Apache*. Namun, belakangan ini PHP juga dapat bekerja dengan *web server*, seperti PWS (*Personal Web Server*), IIS (*Internet Information System*) dan *Xitami*.

PHP bersifat *free* (bebas dipakai), artinya tidak perlu membayar apapun untuk menggunakan perangkat lunak ini. Keunggulan PHP dilihat dari sifatnya *server - side*, antara lain :

1. Tidak diperlukan kontabilitas *browser* atau harus menggunakan *browser* tertentu, karena *server* yang akan mengerjakan skrip PHP. Hasil yang dikirimkan kembali ke *browser* umumnya bersifat teks atau gambar sehingga pasti dikenal oleh *browser* apapun.
2. Dapat memanfaatkan sumber - sumber aplikasi yang dimiliki oleh *server*, misalnya koneksi ke *database*.

2.3.1 Struktur Skrip PHP

Pada umumnya skrip PHP berada dalam kode HTML. Skrip PHP yang berada di dalam kode HTML dapat disimpan dalam ekstensi php. Skrip PHP yang berada di dalam kode XML dapat disimpan juga dalam ekstensi php.

Delimiter merupakan karakter atau kumpulan karakter yang membedakan antara skrip dengan teks biasa dalam HTML. Untuk PHP, *delimiter* yang digunakan adalah sebagai berikut :

1. Untuk dokumen SGML / HTML :

<? skrip PHP ?>

2. Untuk dokumen XML :

```
<? php skrip PHP ?>
```

3. Untuk yang tidak mendukung PHP :

```
<script language="php">
```

```
skrip php
```

```
</script/>
```

4. Dapat juga menggunakan *delimiter* ASP :

```
<% skrip PHP %>
```

jika menggunakan *delimiter* ASP, maka harus mengubah *file* php.ini, yaitu pada bagian *asp_tags* nilainya harus *on*.

2.3.2 Sintaks Dalam PHP

2.3.2.1 Operator

Operator merupakan simbol – simbol yang digunakan untuk melakukan operasi – operasi matematis ataupun operasi *string*. Beberapa operator yang dikenal dalam PHP antara lain :

1. Operator Aritmatika.

Operator Aritmatika adalah operator yang digunakan dalam operasi dasar matematika. Operator – operator yang termasuk operator aritmatika dapat dilihat pada Tabel 2.2 berikut ini.



Tabel 2.2 Operator Aritmatika

Operator	Arti
+	Penambahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus

2. Operator Logika.

Operator logika digunakan untuk membandingkan dua atau lebih pernyataan dan menghasilkan nilai *true* atau *false*. Operator logika sangat berguna untuk menentukan pilihan apabila ada dua atau lebih kondisi yang berbeda. Operator – operator yang termasuk operator logika dapat dilihat pada Tabel 2.3 berikut ini.

Tabel 2.3 Operator Logika

Operator	Arti
And	Operator dan
Or	Operator atau
Xor	Operator atau eksklusif
!	Operator tidak/bukan (<i>not</i>)
&&	Operator dan
	Operator atau

3. Operator Perbandingan.

Operator perbandingan adalah operator yang digunakan untuk membandingkan dua atau lebih nilai, variabel, atau pernyataan dan

menghasilkan nilai. Operator – operator pembanding dapat dilihat pada Tabel 2.4 berikut ini.

Tabel 2.4 Operator Pembanding

Operator	Nama
==	Sama
!=	Tidak sama
<	Lebih kecil
>	Lebih besar
<=	Lebih kecil atau sama dengan
>=	Lebih besar atau sama dengan

2.3.2.2 Struktur Kontrol

Dalam pembuatan sebuah aplikasi sangat jarang eksekusi sebuah skrip berjalan lurus dari awal hingga akhir, biasanya ditemui beberapa kondisi yang mengharuskan alur program bercabang. Struktur kontrol digunakan untuk membantu pengambilan keputusan mengenai alur program mana yang dipilih. Struktur kontrol yang disediakan PHP antara lain adalah sebagai berikut :

1. *If*.

struktur kontrol *if* merupakan struktur kontrol paling lazim dimiliki oleh semua bahasa pemrograman. Struktur kontrol ini digunakan untuk mengambil keputusan apabila ada sebuah kondisi yang ditemui. Jika kondisi tersebut bernilai *true*, maka pernyataan di dalam struktur *if* akan dijalankan. Sebaliknya jika bernilai *false*, pernyataan di dalam struktur *if* akan diabaikan dan langsung menuju ke pernyataan sesudahnya. Sintaks penggunaan *if* adalah sebagai berikut :

```

if (kondisi)
    pernyataan;

```

Jika pernyataan yang akan dijalankan terdiri dari dua baris atau lebih maka bentuknya adalah sebagai berikut :

```

if (kondisi){
    pernyataan;
    pernyataan;
}

```

2. *If...Else.*

Struktur *If...Else* hampir sama dengan *if*, tetapi struktur ini menyediakan pilihan untuk diambil jika sebuah kondisi bernilai *false*. Pilihan tersebut disediakan dalam pernyataan *else*. Jadi jika kondisi bernilai *true*, maka pernyataan di bawah *if* akan dieksekusi, tetapi jika bernilai *false*, pernyataan di bawah *else* akan dieksekusi. Sintaks penggunaan *If...Else* adalah sebagai berikut :

```

If (kondisi)
{
    Pernyataan1;
}else{
    pernyataan2;
}

```

3. *If ... ElseIf...Else.*

Struktur ini memberikan lebih banyak kemungkinan kondisi. Sintaksnya adalah sebagai berikut :

```
If (kondisi1)
{
    Pernyataan1;
} Elseif (kondisi2) {
    Pernyataan2;
    ...
} Elseif(kondisiN) {
    PernyataanN;
} else {
    Pernyataan;
}
```

4. *Switch*.

Struktur *switch* dapat dikatakan sebagai pengganti struktur *if...Elseif...Else*, terutama jika pilihan yang tersedia banyak dan masing – masing pilihan berupa suatu nilai yang pasti. Sintaks penggunaan *switch* sebagai berikut :

```
Switch (variabel) {
    Case nilai1:
        Pernyataan1;
        Break;
    Case nilai2:
```

```
Pernyataan2;
```

```
Break;
```

```
Case nilaiN:
```

```
PernyataanN;
```

```
Break;
```

```
[default:
```

```
pernyataan]
```

```
}
```

5. *While*.

Struktur *while* digunakan untuk mengulang – ulang sebuah blok pernyataan selama suatu kondisi terpenuhi atau bernilai *true*. Kondisi tersebut akan dievaluasi di bagian awal blok pernyataan, jadi jika kondisi terpenuhi, maka blok pernyataan tersebut akan diabaikan. Sintaks penggunaan *while* adalah sebagai berikut :

```
While (kondisi) {
    Pernyataan;
}
```

6. *Do ... while*.

Struktur *do .. while* hampir sama dengan *while*, hanya saja evaluasi kondisi diberikan pada akhir blok pernyataan. Oleh karena itu, blok pernyataan pasti akan dieksekusi terlebih dahulu dan jika pada akhir blok kondisi yang diberikan tidak terpenuhi, maka perulangan dihentikan. Tetapi jika

kondisi terpenuhi maka perulangan akan diteruskan. Sintaks penggunaan *do ... while* adalah sebagai berikut :

```
Do {
    Pernyataan;
} while kondisi;
```

7. *For*.

Struktur *for* digunakan untuk melakukan perulangan (*iterasi*) dengan batas - batas jumlah perulangan yang telah diketahui. Sintaks penggunaan struktur *for* adalah sebagai berikut :

```
For (awal; akhir; langkah) {
    Pernyataan;
}
```

8. *Break*.

Struktur *break* digunakan untuk menghentikan sebuah perulangan, baik pada struktur *for* maupun *while*. Sintaks penggunaan struktur *break* dapat dilihat pada contoh struktur pada *switch*.

2.3.2.3 Fungsi

Fungsi merupakan alat bantu pemrograman yang mempermudah dalam mengerjakan suatu tugas tertentu. Sebuah fungsi merupakan rangkaian dari perintah - perintah pemrograman, yang disusun sedemikian rupa sehingga terangkum dalam pemrograman. Dalam PHP terdapat fungsi yang sengaja disediakan dalam PHP dan ada juga fungsi yang dibuat sendiri oleh *user*. Ada beberapa fungsi yang disediakan dalam PHP dan fungsi yang mendasar dan

umum digunakan, yaitu fungsi *string*. Fungsi *string* merupakan fungsi yang digunakan untuk mengolah data yang berupa *string*. Beberapa fungsi *string* antara lain :

1. Fungsi *Echo()* dan *Print()*.

Fungsi ini berguna untuk menampilkan suatu *string* atau teks ke *browser*.

Sintaksnya sebagai berikut :

```
Echo(string)
```

```
Print(string)
```

2. Fungsi *Strlen()*.

Fungsi ini digunakan untuk mengukur panjang karakter sebuah *string*.

Sintaksnya sebagai berikut :

```
Strlen(string)
```

Sebuah spasi akan dihitung sebagai sebuah karakter.

3. Fungsi *Substr()*.

Fungsi ini digunakan untuk mengambil suatu bagian sebuah *string*.

Sintaksnya sebagai berikut :

```
Substr(string,mulai[,panjang])
```

String akan diambil dari karakter ke - mulai sejumlah panjang.

2.3.2.4 Penanganan *Session*

PHP menyediakan fungsi *session* untuk penyimpanan suatu data agar nilainya tidak hilang dalam suatu *session*. Fungsi-fungsi *session* antara lain :

1. Fungsi *Session_start()*.

Fungsi ini digunakan untuk memulai *session*. Sintaksnya adalah sebagai berikut :

```
session_start()
```

2. Fungsi *Session_destroy()*.

Fungsi ini digunakan untuk mengakhiri *session*. Sintaksnya adalah sebagai berikut :

```
session_destroy()
```

3. Fungsi *Session_name()*.

Fungsi ini digunakan untuk mengambil atau menentukan nama sebuah *session*. Sintaksnya adalah sebagai berikut :

```
session_name([nama])
```

Jika argumen nama tidak disertakan, fungsi digunakan untuk mengambil nama sebuah *session*. Sebaliknya jika argumen nama disertakan, fungsi digunakan untuk memberi nama *session*.

4. Fungsi *Session_module_name()*.

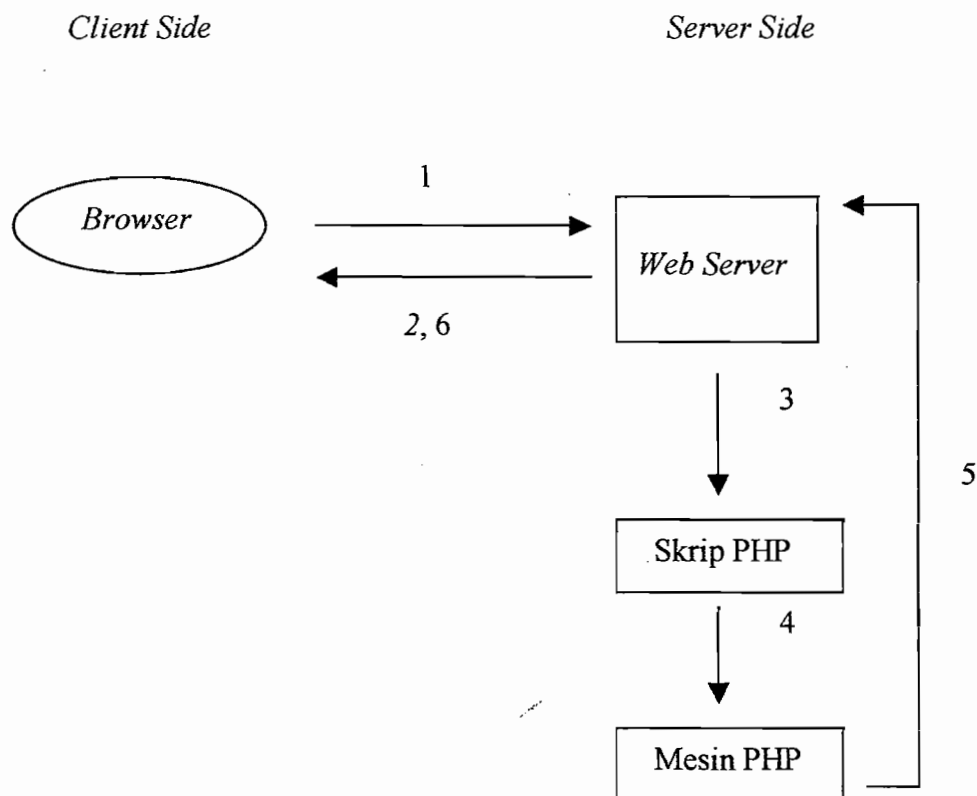
Fungsi ini digunakan untuk mengambil atau menentukan nama sebuah modul atau *session*. Sintaksnya adalah sebagai berikut :

```
session_module_name([modul])
```

Jika argumen modul tidak disertakan, fungsi digunakan untuk mengambil nama sebuah modul *session*. Sebaliknya jika argumen modul disertakan, fungsi digunakan untuk memberi nama pada *session*.

2.3.3 Konsep kerja PHP

Konsep kerja PHP dapat dilihat pada Gambar 2.4 berikut ini.



Gambar 2.4 Konsep Kerja PHP

Keterangan Gambar :

1. *Browser* meminta (*request*) sebuah halaman PHP kepada *web server*.
2. *Server* akan memecah halaman PHP. Bagian PHP (berupa kode HTML), akan langsung dikirim ke *browser*, sedangkan bagian PHP (berupa kode PHP) yaitu skrip PHP akan dikirim ke mesin PHP.
3. *Web server* menemukan skrip PHP.
4. Skrip PHP dikirim ke mesin PHP untuk diproses.
5. Setelah data diolah oleh mesin PHP, maka hasilnya dikirim ke *web server*.

6. Dari *web* server data tersebut akan dikirim ke *browser* sebagai *output* sesuai dengan permintaan.

2.3.4 PHP Membuat Halaman XML

Langkah – langkah yang perlu diperhatikan dalam PHP membuat halaman XML yaitu :

1. Mengaktifkan *extension = php_domxml.dll* pada *file* *php.ini*.

Dengan mengaktifkan *extension = php_domxml.dll* maka *user* dapat membuat skrip XML di dalam PHP sehingga dihasilkan dokumen XML.

2. Menuliskan skrip PHP untuk menghasilkan dokumen XML di dalam *Dreamweaver MX*.
3. Menyimpan skrip yang telah dibuat ke dalam *file* berekstensi PHP.

2.3.4.1 PHP Membuat Halaman XML Secara Dasar

Contoh mengenai PHP membuat halaman XML adalah sebagai berikut :

```

<?php

// Data XML
$xml_string = "<?xml version='1.0'?>
<sentence>What a wonderful profusion of colors and smells in
the market
  <vegetable color='green'>cabbages</vegetable>,
  <vegetable color='red'>tomatoes</vegetable>,
  <fruit color='green'>apples</fruit>,
  <vegetable color='purple'>aubergines</vegetable>,
  <fruit color='yellow'>bananas</fruit>
</sentence>";

// Membuat objek DOM dari data XML
if(!$doc = xmldoc($xml_string))
{
  die("Error parsing XML");
}

// Memulai sebuah root
$root = $doc->root();

// Membuat root untuk children
$children = $root->children();

// Iterasi untuk semua daftar children
foreach ($children as $child)
{
  // Jika <vegetable> adalah elemen
  if ($child->tagname == "vegetable")
  {
    // Ke statement di bawahnya untuk level selanjutnya
    // Menemukan text node
    $text = $child->children();
    // Print isi dari text node
    echo "Found: " . $text[0]->content . "<br>";
  }
}

?>

```

Mengenai *listing* program pada contoh dapat dituliskan langkah – langkah pembuatan program sebagai berikut :

1. Menuliskan versi xml yang digunakan, yaitu versi 1.0.
2. Membuat objek xml, kode program yang digunakan adalah sebagai berikut

```

if(!$doc = xmldoc($xml_string))

```

```
{
    die("Error parsing XML");
}
```

Objek xml diset di dalam `$doc`, program akan mengecek jika `(!$doc = xmldoc($xml_string))` maka akan muncul pesan *error*.

3. Membuat *root* dalam dokumen XML, kode program yang digunakan adalah sebagai berikut :

```
$root = $doc->root();
```

Pada *root* yang dibuat, *root* diset di dalam `$root`.

4. Membuat *root* untuk *children*, kode program yang digunakan adalah sebagai berikut :

```
$children = $root->children();
```

Pada *children* yang dibuat, *children* diset di dalam `$children`.

5. Iterasi untuk daftar *children*, kode program yang digunakan adalah sebagai berikut :

```
foreach ($children as $child)
{
    if ($child->tagname == "vegetable")
    {
        $text = $child->children();
        echo "Found: " . $text[0]->content . "<br>";
    }
}
```

Pada contoh *listing* program contoh, jika skrip program di *run* maka akan menghasilkan *output* sebagai berikut :

Found: cabbages

Found: tomatoes

Found: aubergines

2.3.4.2 PHP Membuat Halaman XML dari MySQL

PHP dapat membuat halaman XML dengan mengambil data yang ada di dalam MySQL. Contoh *listing* program PHP membuat halaman XML yang mengambil data dari MySQL adalah sebagai berikut :

```
<?php

// Membuat DomDocument object
$doc = new_xmldoc("1.0");

// Menambah root node
$root = $doc->add_root("collection");

// query database untuk record - record
$connection = mysql_connect("localhost", "us8749", "secret")
or die ("Unable to
    connect!");
mysql_select_db("db633") or die ("Unable to select
database!");
$query = "SELECT id, title, author, price FROM books";

$result = mysql_query($query) or die ("Error in query: $query.
" . mysql_error());

// Iterasi seluruh resultset
while($row = mysql_fetch_object($result))
{
    $record = $root->new_child("record", "");
    $record->set_attribute("id", $row->id);

    $record->new_child("title", $row->title);
    $record->new_child("author", $row->author);
    $record->new_child("price", $row->price);
}

// Menutup connection
mysql_close($connection);

// dump tree sebagai string
echo $doc->dumplib();
?>
```

Mengenai *listing* program pada contoh dapat dituliskan langkah – langkah pembuatan program sebagai berikut :

1. Membuat *DomDocument object*, kode program yang digunakan adalah sebagai berikut :

```
$doc = new_xmldoc("1.0");
```

2. Menambah *root node*, kode program yang digunakan adalah sebagai berikut :

```
$root = $doc->add_root("collection");
```

3. Membuat *query database* untuk *record – record*.

```
$connection = mysql_connect("localhost", "us8749",
"secret") or die ("Unable to connect!");
mysql_select_db("db633") or die ("Unable to select
database!");
$query = "SELECT id, title, author, price FROM books";
$result = mysql_query($query) or die ("Error in query:
$query. " . mysql_error());
```

Pada pembuatan *query database*, yang pertama kali diperhatikan masalah *connection*. Kemudian dilanjutkan dengan perintah *query database* dari tiap *record*. Pada *listing* program contoh, *query* diset di dalam *\$query* dan hasilnya diset di dalam *\$result*.

4. Membuat Iterasi seluruh *resultset*, kode program yang digunakan adalah sebagai berikut :

```
while($row = mysql_fetch_object($result))
{
    $record = $root->new_child("record", "");
    $record->set_attribute("id", $row->id);
    $record->new_child("title", $row->title);
    $record->new_child("author", $row->author);
    $record->new_child("price", $row->price);
}
```

5. Menutup *connection*, kode program yang digunakan adalah sebagai berikut

:

```
mysql_close($connection);
```

6. *Dump tree* sebagai *string*. *Dump* ini digunakan untuk menampilkan data.

Kode program yang digunakan adalah sebagai berikut :

```
echo $doc->dumppem();
```

2.4 MySQL

MySQL adalah perangkat lunak pengolah *database*. Badan yang membuat MySQL adalah MySQL AB, Swedia. MySQL merupakan perangkat lunak yang bersifat *open source*. Sesuai dengan namanya, bahasa standar yang digunakan oleh MySQL adalah SQL. SQL merupakan singkatan dari *Structured Query Language*. SQL merupakan bahasa standar untuk pengolahan *database*.

2.4.1 Sintaks MySQL

Sintaks - sintaks dalam bahasa MySQL antara lain :

1. *Create Database.*

Perintah ini digunakan untuk membuat suatu *database*. Sintaksnya sebagai berikut :

```
create database[if not exist] namadatabase
```

Jika pernyataan *If Not Exist* tidak disertakan, maka apabila ada nama yang *database* yang sama dengan nama *database* di server, akan terjadi *error*.

2. *Create Table.*

Perintah ini digunakan untuk membuat tabel. Sintaksnya adalah sebagai berikut :

create table namatabel

(

struktur

);

3. *Drop Database.*

Perintah ini digunakan untuk menghapus sebuah *database*. Sintaksnya adalah sebagai berikut :

drop database namadatabase

4. *Update Record.*

Update Record untuk memodifikasi atau mengubah isi *record* tertentu dengan menggunakan perintah sebagai berikut:

update namatabel set kolom1=nilaibaru1, kolom2=nilaibaru2 ... where kriteria;

5. *Insert.*

Perintah ini digunakan untuk menyisipkan suatu data ke dalam tabel. Data yang disisipkan dapat berupa data yang diambil dari tabel lain, ataupun data yang berupa nilai - nilai tertentu yang disebutkan secara *eksplisit*.

Untuk data yang diambil dari tabel lain, sintaksnya sebagai berikut :

Insert [into] namatabel [(namafiled...)] perintah select

Untuk data yang disebutkan secara *eksplisit*, sintaksnya sebagai berikut :

Insert[into] namatabel [(namafiled...)] VALUES ekspresi

7. *Select.*

Perintah ini digunakan untuk mengambil data dari suatu tabel. Sintaksnya sebagai berikut :

```
select {*:namafield} from namatabel
[into tabeltujuan] [where kondisi]
```

8. *Delete.*

Perintah ini digunakan untuk menghapus sebuah *record* dari tabel. Sintaksnya sebagai berikut :

```
Delete from namatabel where kondisi
```

9. *Use.*

Perintah ini digunakan untuk memilih *database* yang akan digunakan. Sintaksnya sebagai berikut :

```
use namatabel
```

2.4.2 Tipe Data MySQL

MySQL memiliki beberapa tipe data. Beberapa tipe data yang sering dipakai dapat dilihat pada Tabel 2.5.

Tabel 2.5 Tipe – tipe data dalam MySQL
(Anon Kuncoro Widigdo, Dasar pemrograman PHP dan MySQL, ilmukomputer.com, hlm 9)

Tipe data	Keterangan
INT(M) [UNSIGNED]	Angka -2147483648 s/d 2147483647
FLOAT(M,D)	Angka pecahan
DATE	Tanggal Format : YYYY-MM-DD
DATETIME	Tanggal dan Waktu Format : YYYY-MM-DD HH:MM:SS
CHAR(M)	String dengan panjang tetap sesuai dengan yang ditentukan. Panjangnya 1-255 karakter
VARCHAR(M)	String dengan panjang yang berubah-ubah sesuai dengan yang disimpan saat itu. Panjangnya 1 – 255 karakter
BLOB	Teks dengan panjang maksimum 65535 karakter
LONGBLOB	Teks dengan panjang maksimum 4294967295 karakter

2.4.3 Kelebihan MySQL

Kelebihan MySQL antara lain adalah :

1. *Free* (bebas dipakai), artinya tidak perlu membayar apapun untuk menggunakan perangkat lunak ini.
2. Cocok dalam pengaksesan *database* pada *internet*, karena kecepatan MySQL.

2.5 Visual Basic

2.5.1 Pengenalan Visual Basic

Visual Basic adalah bahasa pemrograman tingkat tinggi yang merupakan pengembangan dari bahasa BASIC (*Beginners' Allpurpose Symbolic Instruction Code*) versi DOS.

Perbedaan antara Visual Basic dengan bahasa BASIC adalah pemrograman BASIC masih berorientasi pada *text* dan program dieksekusi secara berurutan. Untuk itu bahasa BASIC disebut sebagai *Interpreter*. Sedangkan dalam Visual Basic dengan lingkungan grafiknya, pemrograman berorientasi *object* dan sudah merupakan *compiler*. Visual Basic program terdiri dari banyak sub program atau prosedur, dimana setiap prosedur mempunyai kode tersendiri dan dapat dieksekusi sendiri dan pada saat yang bersamaan dapat digabungkan menjadi satu.

Pengembangan program dalam Visual Basic dapat dibuat hanya dengan beberapa langkah mudah, terutama untuk pembuatan *interface* dengan *user*.

Visual Basic mempunyai beberapa komponen sebagai berikut :

1. *Full Screen* yang lengkap.

2. Lingkungan eksekusi (*runtime environment*) yang memungkinkan *user* melihat hasil dari program dengan hanya satu kali *click*.
3. *Project Manager* yang memungkinkan pemakai membuat *multiple - file Visual Basic Windows Applications*.
4. *Interactive Testing Platform* yang memungkinkan *user* melacak kesalahan (*bug*) dari program.
5. *Visual Tools* yang memungkinkan *user* mengatur *control* dan *icon* yang membuat program berfungsi normal.

2.5.2. Object Microsoft DOM dalam Visual Basic

Object microsoft DOM dalam Visual Basic merupakan komponen yang digunakan untuk mengkoneksikan dokumen XML ke dalam Visual Basic. Komponen *microsoft* DOM ini biasanya sudah tersedia di aplikasi Visual basic dari *microsoft*, tetapi jika komponen *microsoft* DOM belum tersedia maka *user* dapat *download* terlebih dahulu untuk *file* DOM ini. Komponen *microsoft* DOM dapat di *download* secara gratis pada alamat <http://msdn.microsoft.com>. Contoh *load* sebuah dokumen XML ke dalam *object DOMDocument*, yaitu :

```
Private Sub LoadXMLFile_Click()
Dim ObjXMLDoc As DOMDocument

    'Membuat DOMDocument
    Set ObjXmlDoc = New Document
    'load dokumen XML, dengan metode load sebagai berikut
    'ini
    objXmlDoc.async = False
    objXmlDoc.Load ("http ://Evy/Virus_komputer.xml")
End Sub
```

Jika pada aplikasi Visual Basic telah ada *object Microsoft* DOM maka aplikasi Visual Basic sudah dapat menerima *file* XML. Dalam *menload* dokumen

XML ke dalam Visual Basic diperlukan pembuatan dokumen DOM terlebih dahulu, kode program untuk set *new document* adalah sebagai berikut :

```
Set ObjXmlDoc = New Document
```

Selanjutnya untuk *menload* dokumen XML dapat digunakan metode *load* sebagai berikut :

```
objXmlDoc.async = False
objXmlDoc.Load ("http : //Evy/Virus_komputer.xml")
```

artinya bahwa aplikasi visual Basic akan mengambil data dari alamat Evy dan *file* yang akan *di-load* adalah Virus_komputer.xml. *File* ini bisa saja berekstensi lain seperti php ataupun asp, karena mungkin saja XML berada dalam php atau asp yang menghasilkan dokumen XML.

2.6 Informasi Virus Komputer

Pada awalnya program "VIRUS" diciptakan untuk melindungi program yang dibuat oleh *programmer* agar karyanya tidak dibajak oleh orang lain (hanya memberi pelajaran). Tetapi hal ini kemudian berkembang menjadi suatu hal yang menimbulkan dampak negatif ketika orang mulai menggunakan virus komputer bukan lagi untuk melindungi karyanya tetapi sudah mulai melakukan tindakan *destruktif* yang bertujuan menghancurkan lawan atau musuh.

Sampai saat sekarang berkembang model - model virus yang mampu melakukan kegiatan mata - mata, pemhancuran dokumen, pemalsuan dokumen, dll. Beberapa contoh virus komputer, antara lain : *worm, sobig.c, sobig.d, sobig.f, mellisa, blaster, macro, worm_klez, bubble boy, minizip, code red, my party, mypics, babylonia, trojans, march, i love u*, dll.

2.6.1 Definisi Virus Komputer

¹Virus komputer didefinisikan sebagai :

1. Sebuah program yang mampu melakukan duplikasi dirinya didalam sistem komputer yang dimasukinya, dan memiliki potensi besar untuk melakukan manipulasi terhadap sistem tersebut.
2. Virus adalah sebuah kode komputer yang dapat dijalankan dan berukuran kecil dengan kemampuan memperbanyak diri, baik dengan cara menempelkan sebagian atau seluruh *file* atau aplikasi program, dan mengakibatkan komputer *user* melakukan hal-hal yang tidak diinginkan *user*.

2.6.2 Karakteristik Virus Komputer

Seiring kemajuan jaman, virus komputer dalam perkembangannya juga mengalami kemajuan sehingga memiliki efek yang semakin mengerikan dengan sifat merusaknya.

Beberapa karakteristik virus komputer itu adalah:

1. Memiliki kemampuan untuk melakukan penularan pada *file - file* eksekusi program secara *ilegal*.
2. Dapat bergerak berpindah-pindah dari satu komputer yang satu ke komputer yang lain dengan menggunakan media yang dimiliki komputer itu sendiri, seperti : *floppy disk*, *hardisk*, dan *cd room*. Bahkan juga dapat melalui *e-mail* atau melalui *file* hasil *download* melalui *internet*.

¹ Wijang Widhiarso, S.Kom, *Makalah seminar tentang virus*, STMIK MDP Palembang 9 November 2002, hal 2.

3. Dapat melakukan fungsi *illegal*, seperti mengacaukan waktu, merusak, sukar dilacak, dan lain-lain.

2.6.3 Jenis Virus Komputer

Saat ini banyak jenis variasi virus yang beredar, kebanyakan di antaranya dapat dikelompokkan menjadi enam kategori umum. Enam kelompok virus komputer tersebut adalah sebagai berikut :

1. **Virus *boot - sector*** : menggantikan atau memasukkan dirinya ke dalam *boot - sector* sebuah area pada *hard drive* (atau jenis *disk* lainnya) yang akan diakses pertama kali saat komputer dinyalakan. Virus jenis ini dapat menghalangi komputer untuk melakukan *booting* dari *hard disk*.
2. **Virus *file*** : menginfeksi aplikasi. Virus ini melakukan eksekusi untuk menyebarkan dirinya pada aplikasi dan dokumen yang terkait dengannya saat *file* yang terinfeksi dibuka atau dijalankan. |
3. **Virus *makro*** : ditulis dengan menggunakan bahasa pemrograman *makro* yang disederhanakan, dan menginfeksi aplikasi *Microsoft Office*, seperti *Word* dan *Excel*, dan saat ini diperkirakan 75 persen dari jenis virus ini telah tersebar di dunia. Sebuah dokumen yang terinfeksi oleh virus *makro* secara umum akan memodifikasi perintah yang telah ada dan banyak digunakan (seperti perintah "*Save*") untuk memicu penyebaran dirinya saat perintah tersebut dijalankan.
4. **Virus *multipartite*** : menginfeksi baik *file* dan *boot - sector*, sebuah penjahat berganda yang dapat menginfeksi sistem *user* terus menerus sebelum ditangkap oleh *scanner* antivirus.

5. **Virus *polymorphic*** : akan mengubah kode dirinya saat dilewatkan pada mesin yang berbeda. Secara teoritis virus jenis ini lebih susah untuk dapat dideteksi oleh *scanner* antivirus, tetapi dalam kenyataannya virus jenis ini tidak ditulis dengan baik, sehingga mudah untuk diketahui keberadaannya.
6. **Virus *stealth*** : menyembunyikan dirinya dengan membuat *file* yang terinfeksi tampak tidak terinfeksi, tetapi virus jenis ini jarang mampu menghadapi *scanner* antivirus terbaru.

2.6.4 Cara Kerja Virus komputer

Dalam melakukan operasi menyusupkan kode programnya, ada beberapa cara kerja virus komputer itu, yaitu:

1. *Virus Overwriting Target File.*

Menindas *file* program eksekusi sasaran dan menggantinya dengan program virusnya.

2. *Virus Non - Overwriting Target File.*

Kemampuan menyusup tanpa merusak program *file* sasaran, hanya memindahkan lokasi program asli dan menggantikan *header* program file target dengan programnya sendiri.

3. *Virus Resident.*

Melakukan penularan dengan menggunakan memori RAM komputer. Virus akan berada di dalam memori sampai komputer dimatikan.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai analisis dan perancangan sistem yang akan dibangun.

3.1 Sistem yang dikembangkan

Sistem yang akan dikembangkan adalah sistem pengaksesan *database server* ke dalam format dokumen XML, sehingga dapat diakses pada aplikasi yang berbeda dengan memanfaatkan protokol HTTP. Program yang akan dibuat bersifat *client - server*, *database* yang akan diolah adalah *database* virus komputer akan diolah di *server* untuk diubah menjadi format dokumen XML, sehingga dokumen tersebut dapat diakses di *client* pada aplikasi yang berbeda.

3.2 Kebutuhan Sistem

Sistem terdiri dari dua *side*, yaitu *server side* dan *client side*. Teknologi yang digunakan dalam *server side* yaitu menggunakan PHPTriad v2-11 yang dapat di *download* pada <http://sourceforge.net/projects/phptriad>. PHPTriad adalah program yang memudahkan instalasi *Apache Web server*, PHP, basisdata MySQL, dan fasilitas PHPMyAdmin.

PHPTriad v2-11 menyediakan *Apache Web server*, PHP, *database* MySQL, dan PHPMyAdmin dengan spesifikasi sebagai berikut :

1. PHP - 4.0.5
2. *Database* MySQL - 3.23.32
3. *Server Web Apache* - 1.3.14

4. PHPMyAdmin - 2.1.0, digunakan sebagai sistem pengaturan *database* MySQL berbasis GUI (*Graphic User Interface*).

Pada *server side* akan melakukan pengaksesan *database server* dengan memanfaatkan skrip PHP sehingga dihasilkan data dalam bentuk dokumen XML yang dapat diakses pada aplikasi berbeda.

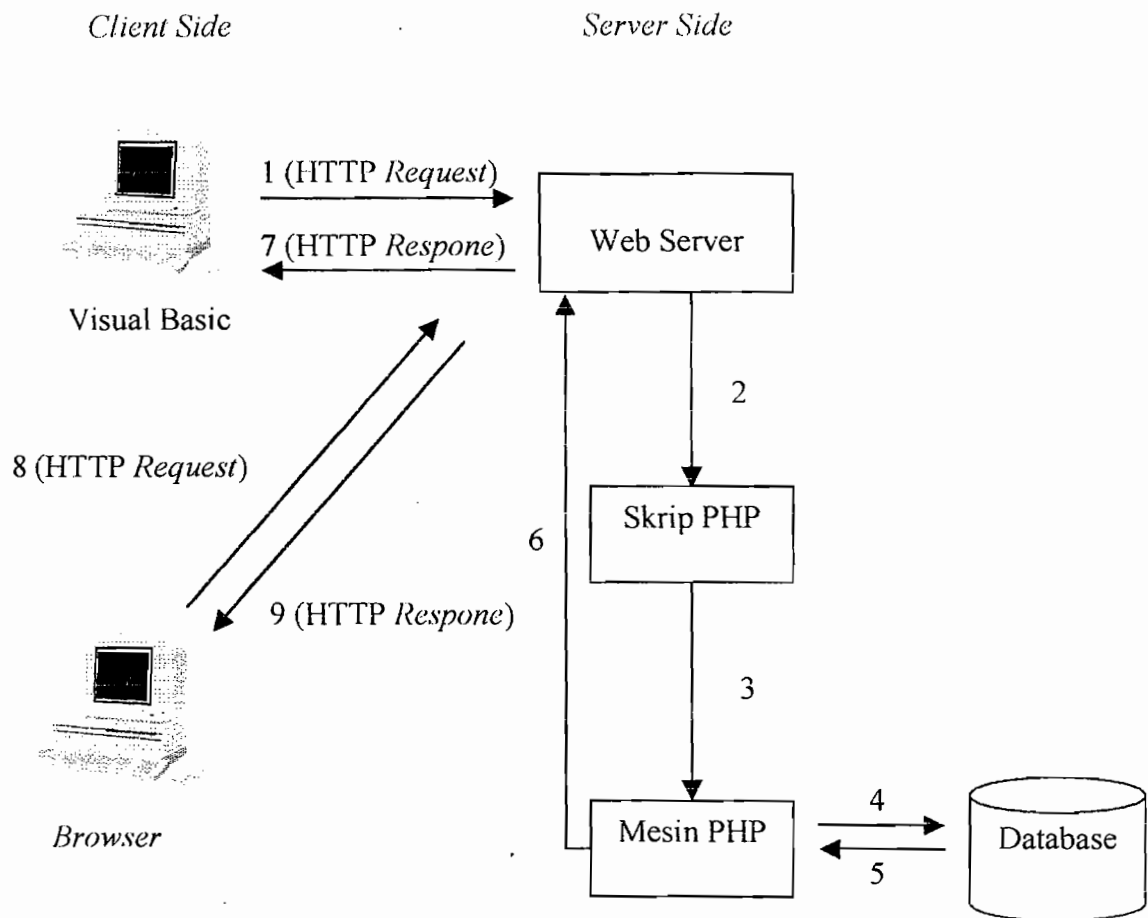
Teknologi *Client Side* menggunakan Visual Basic 6.0, yang digunakan untuk menampilkan informasi virus komputer.

3.3 Analisis Sistem

3.3.1 Konsep Kerja Sistem

Konsep kerja sistem aplikasi XML dalam pengaksesan *database server* akan digambarkan pada Gambar 3.1 beserta dengan penjelasannya.





Gambar 3.1 Konsep Kerja Sistem

Keterangan :

1. *Client Side*, yaitu Visual Basic akan melakukan *request* terhadap sistem, yaitu *request* `xml_evy.php` ke *web server*. Modul yang digunakan untuk membangun *interface* pada *client side* adalah Modul *load* (`Load.frm`), *setup* (`Setup.frm`), *information of virus computer* (`Virus.frm`), *search* (`Search.frm`), *about programmer* (`AboutProgrammer.frm`), *about program* (`Aboutprogram.frm`).
2. *Web server* akan mencari halaman PHP dan menemukan skrip PHP.

3. Skrip PHP akan dikirim ke mesin PHP untuk diproses, skrip PHP yang dikirim adalah skrip dari modul `xml_evy.php`.
4. Karena dalam skrip PHP melibatkan pengaksesan *database*, yaitu *database* virus komputer maka mesin PHP akan melakukan proses berupa permintaan dengan menggunakan SQL *query* ke MySQL *server*.
5. Setelah sampai di MySQL *server* maka permintaan tersebut akan ditulis (*write*) ke dalam *file database* dan akan dilakukan pemrosesan. *File database* akan membaca (*read*) proses tersebut. Hasil dari pembacaan proses tersebut berupa data virus komputer yang akan dikirimkan ke mesin PHP. *Database* virus komputer berisi data – data virus komputer yang dimasukkan oleh administrator dalam bentuk *interface* yang dipanggil di *browser*. Modul yang digunakan untuk *input* data – data virus komputer adalah modul `index.php`, `home.php`, `addvirus.php`, `addtarget.php`, `addsolution.php`, `logout.php`.
6. Skrip PHP yang telah diolah dalam mesin PHP menghasilkan dokumen XML dan akan dikirim ke *web server*.
7. *Web server* akan mengirimkan `xml_evy.php` yang berupa format dokumen XML yang akan di *load* di aplikasi *client*, yaitu Visual Basic.
8. *Server side*, yaitu administrator dapat melakukan *request* terhadap sistem, yaitu *request* `xml_evy.php` ke *web server*.
9. Sistem akan mengulangi langkah 2 sampai 6 sehingga *web server* akan mengirimkan dokumen XML.

3.3.2 DAD (Data Arus Diagram)

²Kesatuan luar merupakan kesatuan (*entity*) di luar sistem, karena di luar proses pengolahan data (sistem informasi). Kesatuan luar merupakan sumber arus data ke sistem informasi serta tujuan penerima arus data hasil dari proses sistem informasi.

Dalam menggambar DAD yang baik, diperlukan adanya pengidentifikasian kesatuan luar dan pengidentifikasian *input output* yang terlibat dalam kesatuan luar. Identifikasi kesatuan luar dan identifikasi *input output* yang terlibat dalam kesatuan luar pada sistem yang akan dibuat adalah sebagai berikut :

1. Identifikasi Kesatuan Luar.

a. Administrator.

b. *User*.

2. Identifikasi *Input* dan *Output* yang terlibat dengan kesatuan luar.

a. Administrator.

Input : Data – data virus komputer, ID admin, nama admin, *password*.

Output: Data virus komputer.

b. *User*.

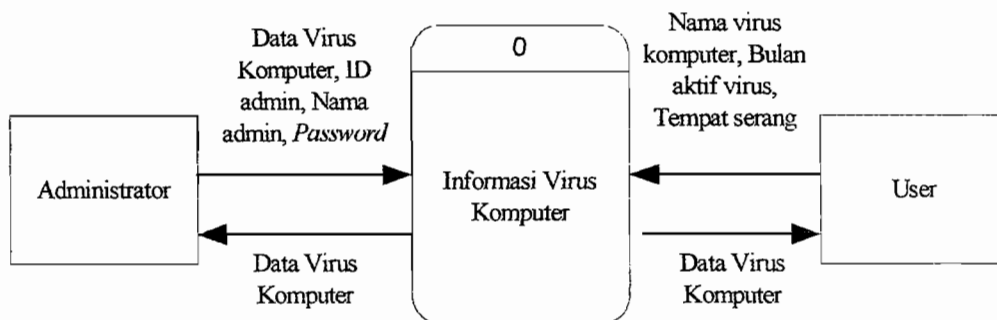
Input : Nama virus komputer, bulan aktif virus dan tempat serang.

Output: Data virus komputer.

² Jogiyanto HM, Akt., MBA, Ph.D., *Analisis & Desain Sistem Informasi : pendekatan terstruktur teori dan praktek aplikasi bisnis*, ANDI Yogyakarta, 2001, hal. 713.

3.3.2.1 Konteks Diagram

Konteks diagram menggambarkan suatu proses yang mewakili proses dari seluruh sistem dan menggambarkan hubungan *input* dan *output* antara sistem dan kesatuan luar. Konteks diagram sistem dapat dilihat pada Gambar 3.2.

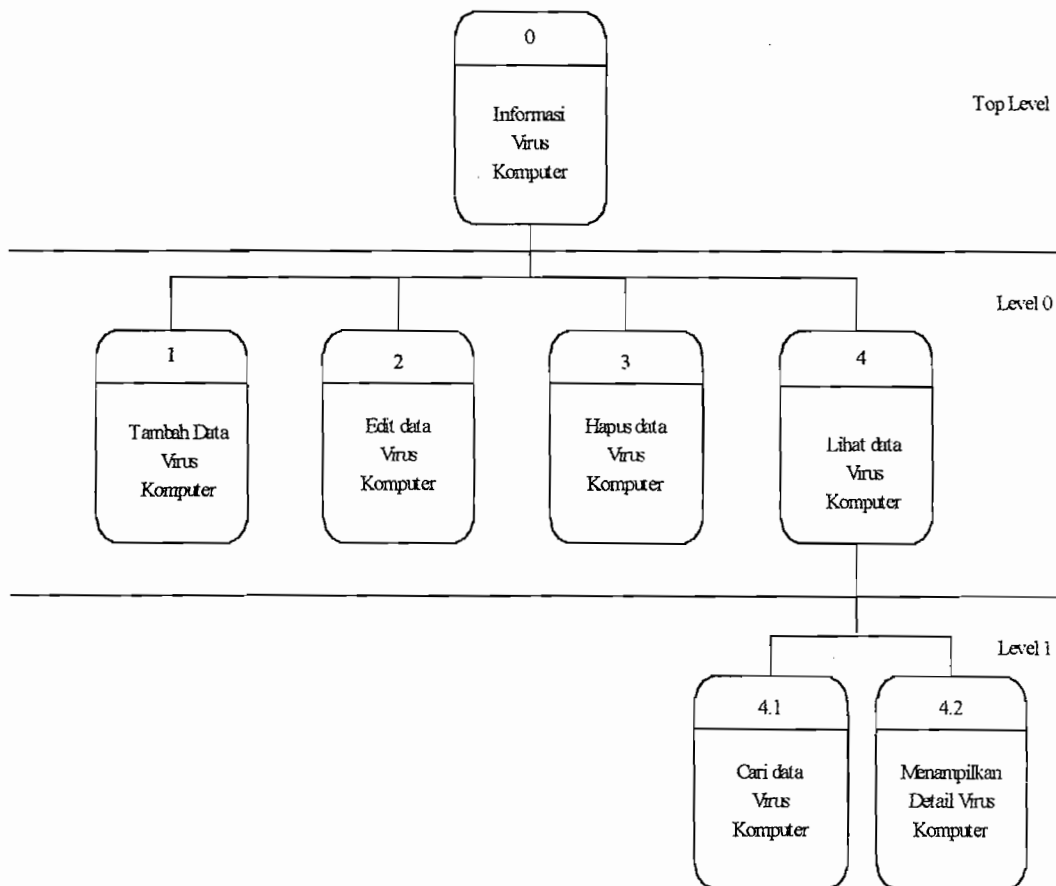


Gambar 3.2 Konteks Diagram Informasi Virus Komputer

Dari Gambar 3.2 dapat dilihat bahwa administrator memasukkan ID admin, Nama admin dan *Password* agar dapat melakukan proses tambah data, *edit* data dan hapus data virus komputer ke dalam sistem dan data – data tersebut akan tertampil di layar yang dapat dilihat oleh administrator pada pihak *server* dalam tampilan untuk administrator dan *user* pada pihak *client* dalam tampilan aplikasi visual basic. Jika *user* melakukan proses pencarian maka *user* akan memasukkan *virus_name*, *active_time* dan *target* ke dalam sistem untuk mendapatkan nama virus komputer sesuai dengan yang dikehendaki *user*.

3.3.2.2 Diagram Berjenjang

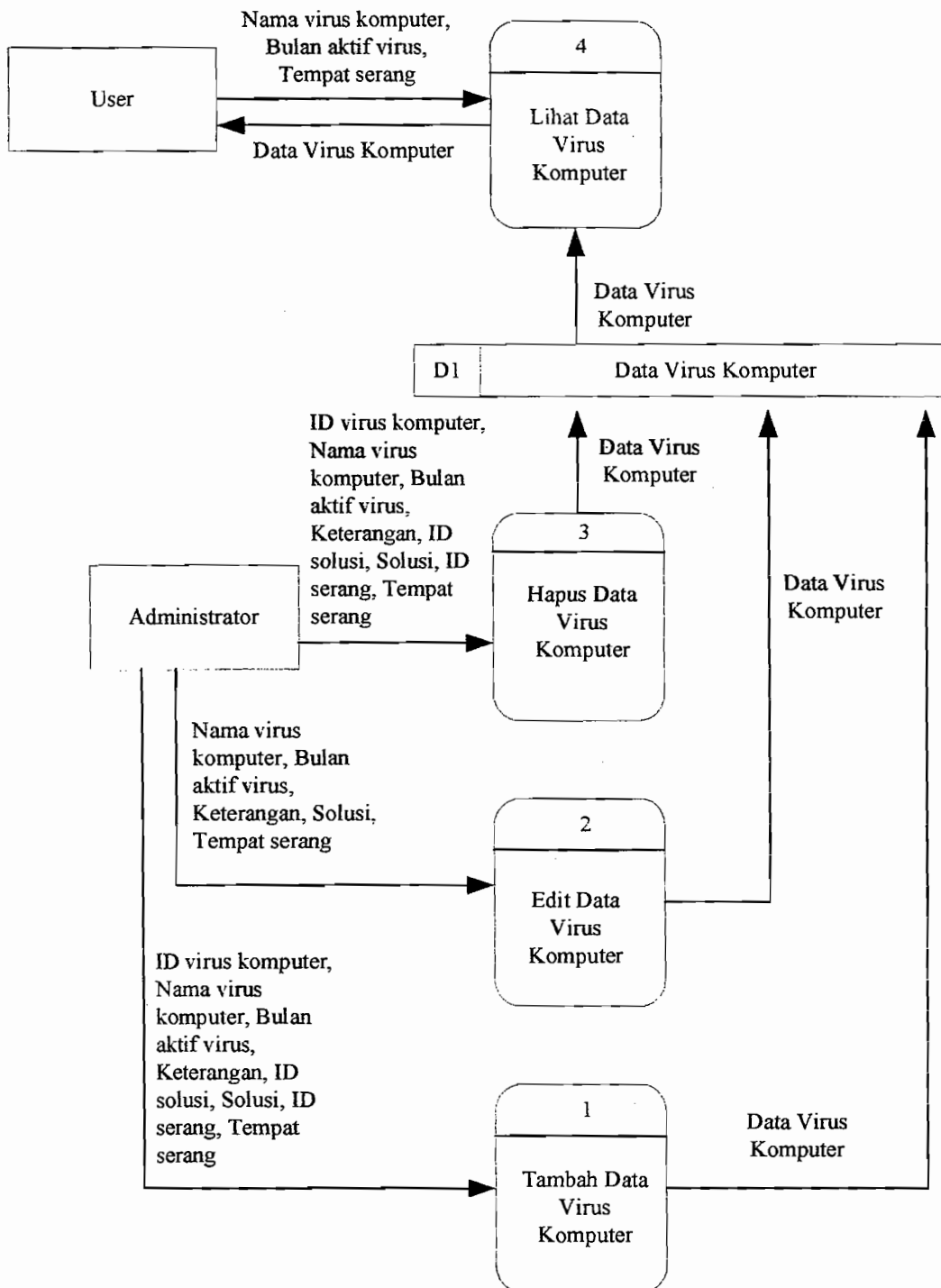
Diagram berjenjang dari sistem dapat dilihat pada Gambar 3.3 berikut ini.



Gambar 3.3 Diagram Berjenjang Sistem

3.3.2.3 DAD Level 0

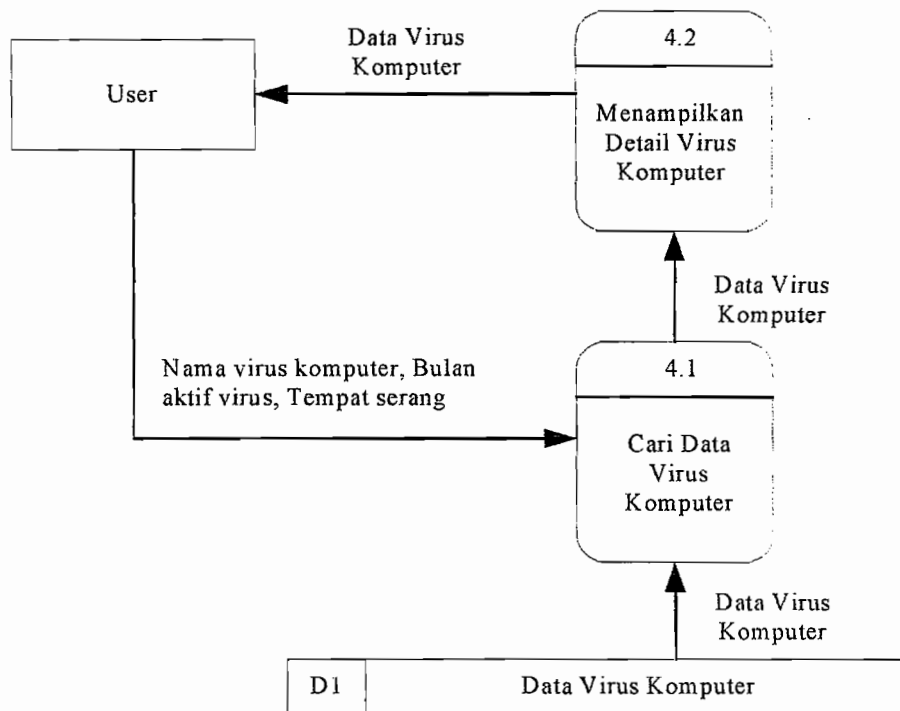
DAD level 0 dari sistem dapat dilihat pada Gambar 3.4 berikut ini.



Gambar 3.4 Gambar DAD Level 0

3.3.2.4 DAD Level 1 (Proses 4)

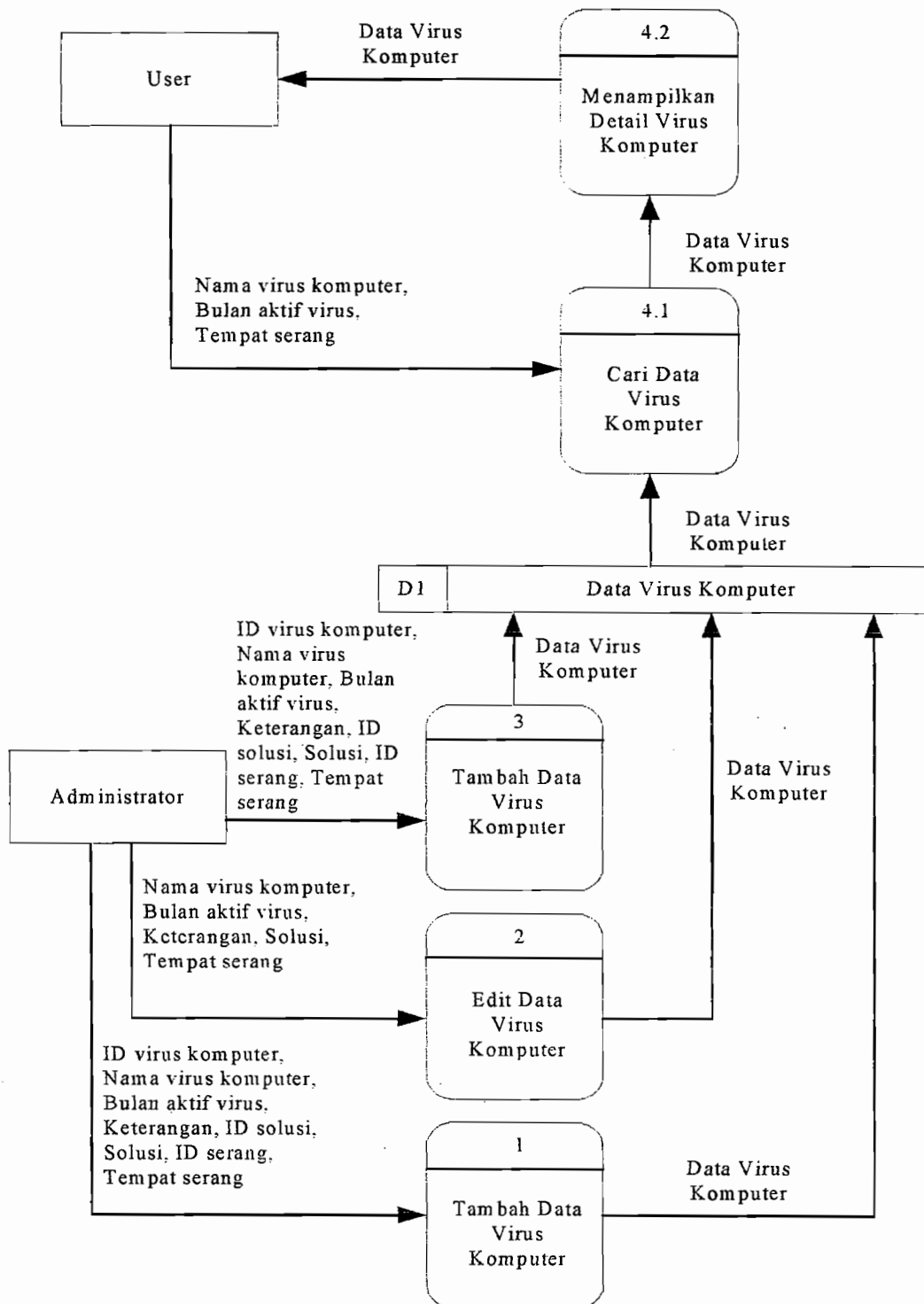
DAD level 1 untuk proses 4 dapat dilihat pada Gambar 3.5 berikut ini.



Gambar 3.5 DAD Level 1 (Proses 4)

3.3.2.5. Diagram Keseluruhan

Diagram keseluruhan dari sistem tampak pada Gambar 3.6 berikut ini.



Gambar 3.6 Diagram Keseluruhan Sistem

3.3.2.6 Kamus Data

³Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan – kebutuhan informasi dari suatu sistem informasi. Kamus data dalam informasi virus komputer adalah sebagai berikut :

1. Nama Arus Data : Nama virus komputer.
 - Nama Alias : -
 - Bentuk Data : Item Data.
 - Arus Data : Administrator – Proses 1, Administrator – Proses 2
 , Administrator – Proses 3, *User* – Proses 4.1.
 - Penjelasan : Berisi nama virus komputer yang akan dicari oleh
user.
 - Periode : Setiap dilakukan proses tambah data,
edit data, hapus data dan caridata.
 - Struktur Data : nama_ virus (*Varchar*).
2. Nama Arus Data : Bulan aktif virus.
 - Nama Alias : -
 - Bentuk Data : Item Data.
 - Arus Data : Administrator – Proses 1, Administrator – Proses 2
 , Administrator – Proses 3, *User* – Proses 4.1.
 - Penjelasan : Berisi bulan aktif virus komputer yang akan dicari
 oleh *user*.

³ Jogiyanto HM, Akt., MBA, Ph.D., *Analisis & Desain Sistem Informasi : pendekatan terstruktur teori dan praktek aplikasi bisnis*, ANDI Yogyakarta, 2001, hal. 725. . .

- Periode : Setiap dilakukan proses tambah data,
edit data, hapus data dan caridata.
- Struktur Data : bulan_aktif_virus (*Varchar*).
3. Nama Arus Data : ID virus komputer.
- Nama Alias : -
- Bentuk Data : Item data.
- Arus Data : Administrator – Proses 1, Administrator – Proses 3.
- Penjelasan : Berisi tentang nomor ID (nomor unik) dari
virus komputer yang berbeda satu dengan
yang lain.
- Periode : Setiap dilakukan tambah dan hapus data
virus komputer.
- Struktur Data : ID_virus (*Bigint*).
4. Nama Arus Data : ID solusi.
- Nama Alias : -
- Bentuk Data : Item data.
- Arus Data : Administrator – Proses 1, Administrator - Proses 3.
- Penjelasan : Berisi tentang nomer ID (nomor unik) dari solusi
yang berbeda satu dengan yang lain.
- Periode : Setiap dilakukan tambah dan hapus solusi.
- Struktur Data : ID_solusi (*Bigint*).
5. Nama Arus Data : Solusi.
- Nama Alias : -

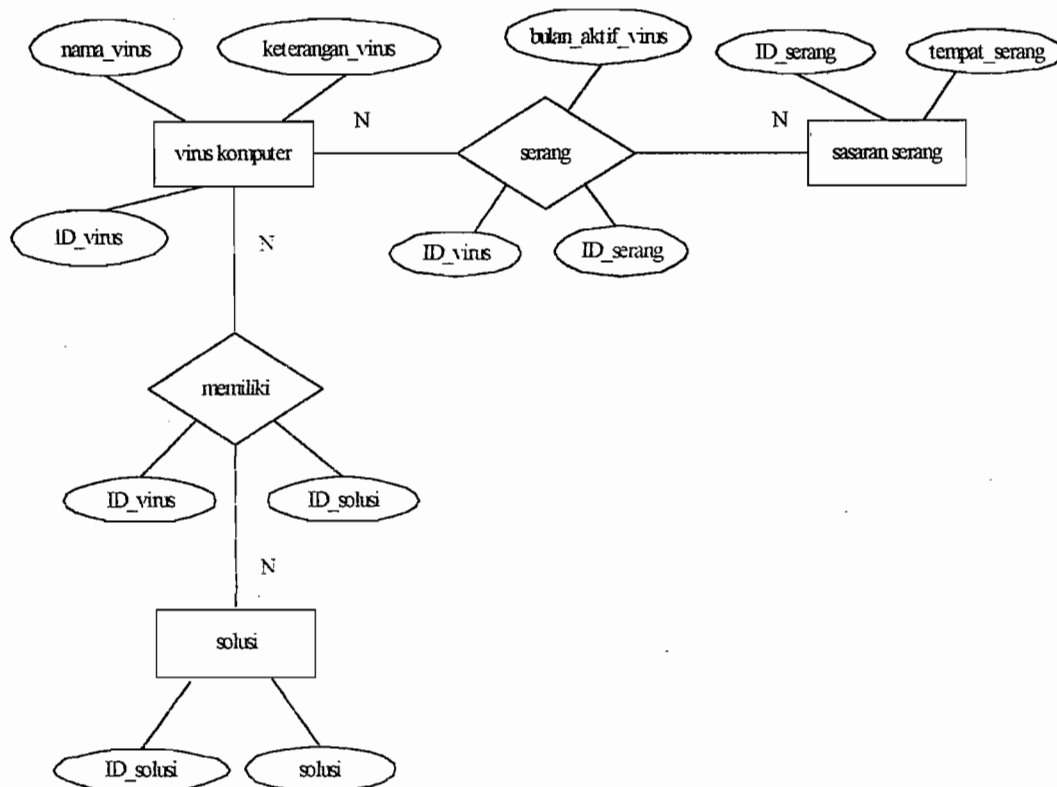
- Bentuk Data : Item data.
- Arus Data : Administrator – Proses 1, Administrator – Proses 2, Administrator – Proses 3.
- Penjelasan : Berisi uraian solusi mengenai solusi penyerangan virus komputer.
- Periode : Setiap dilakukan tambah, *edit* dan hapus data virus komputer.
- Struktur Data : Informasi solusi mengenai virus komputer terdiri beberapa item data :
- ID_solusi (*Bigint*)
- solusi (*Longtext*)
6. Nama Arus Data : ID Serang.
- Nama Alias : -
- Bentuk Data : Item Data.
- Arus Data : Administrator – Proses 1, Administrator – Proses 3.
- Penjelasan : Berisi tentang nomer ID (nomor unik) dari sasaran serang yang berbeda satu dengan yang lainnya.
- Periode : Setiap kali dilakukan proses tambah data dan hapus data virus komputer.
- Struktur Data : ID_serang (*Bigint*).
7. Nama Arus Data : Tempat serang.
- Nama Alias : -

- Bentuk Data : Item Data.
- Arus Data : Administrator – Proses 1, Administrator – Proses 2, Administrator – Proses 3, User – Proses 4.1.
- Penjelasan : Berisi mengenai tempat penyerangan virus komputer dalam menyerang komputer.
- Periode : Setiap dilakukan tambah, *edit*, hapus dan cari data virus komputer.
- Struktur Data : Informasi tempat penyerangan virus komputer terdiri dari beberapa item data :
- ID_serang (*Bigint*)
- tempat_serang (*Varchar*)
8. Nama Arus Data : Keterangan.
- Nama Alias : -
- Bentuk Data : Item Data.
- Arus Data : Administrator – Proses 1, Administrator – Proses 2, Administrator – Proses 3.
- Penjelasan : Berisi keterangan virus komputer.
- Periode : Setiap dilakukan proses tambah, *edit* dan hapus data virus komputer.
- Struktur Data : keterangan (*Longtext*).
9. Nama Arus Data : Data virus komputer.
- Nama Alias : -
- Bentuk Data : Item Data.

Arus Data	: Proses 1 – D1, Proses 2 – D1, Proses 3 – D1, D1 – Proses 4.1, Proses 4.1 – Proses 4.2, Proses – 4.2 – <i>User</i> .
Penjelasan	: Berisi data virus komputer yang terdiri dari ID virus komputer, nama virus komputer, keterangan, ID solusi, solusi, ID serang, tempat serang.
Periode	: Setiap dilakukan proses tambah data virus komputer.
Struktur Data	: Informasi tentang virus komputer beberapa item data : ID_virus (<i>Bigint</i>) nama_virus (<i>Varchar</i>) keterangan(<i>Longtext</i>) ID_solusi (<i>Bigint</i>) solusi (<i>Longtext</i>) ID_serang (<i>Bigint</i>) tempat_serang (<i>Varchar</i>)

3.3.3 ERD (*Entity Relational Diagram*)

ERD dibuat dengan tujuan untuk menggambarkan hubungan antar entitas (*entity*) yang digunakan oleh sistem secara keseluruhan. Bentuk ERD dalam sistem dapat dilihat pada Gambar 3.7.



Gambar 3.7 ERD Sistem

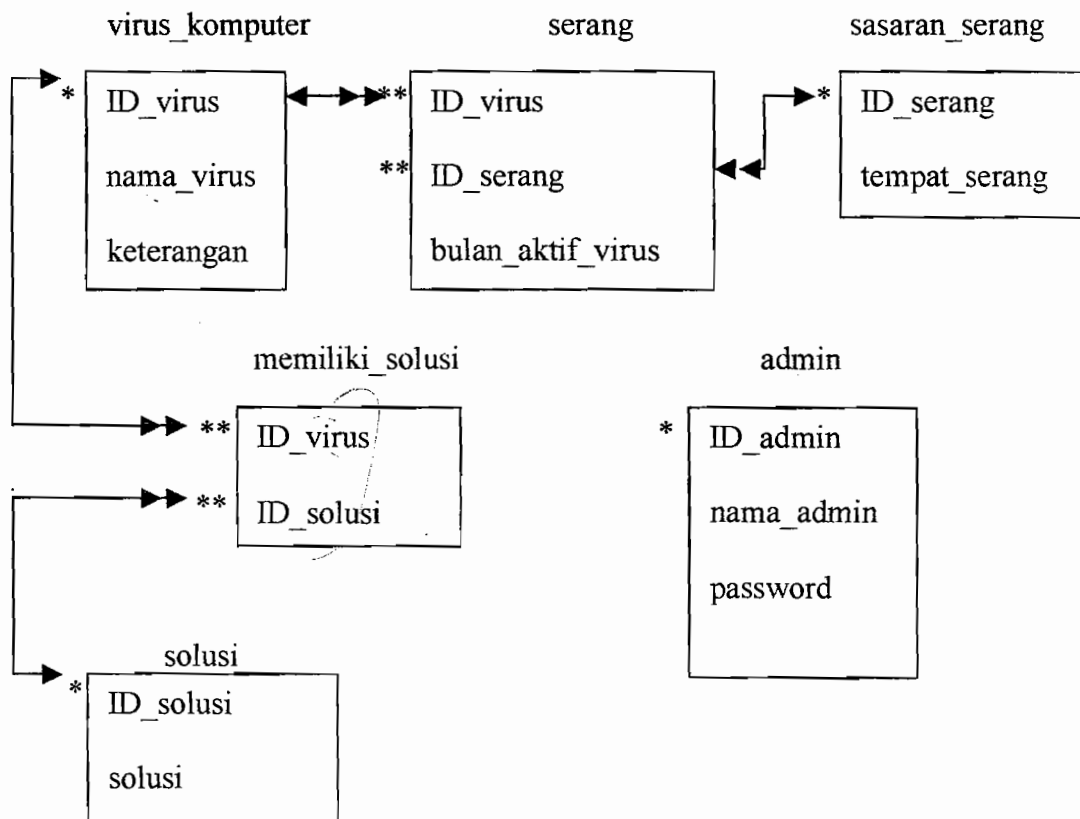
3.4 Perancangan Sistem

3.4.1 Perancangan Basis Data

Dalam rancangan *database* ini akan dirancang suatu *database* yang dibangun dari pemodelan sistem dalam bentuk ERD. *Database* yang dibangun terdiri dari beberapa tabel yang akan menyimpan data - data dari sebuah entitas, sedangkan *record* dalam tabel akan menyimpan nilai atribut - atribut dari suatu entitas.

Tabel – tabel yang digunakan dalam basis data ini adalah tabel *virus_komputer*, *serang*, *sasaran_serang*, *memiliki_solusi*, *solusi* dan *admin*.

Hubungan antar keseluruhan tabel yang digunakan dalam basis data ini dapat dilihat pada Gambar 3.8.



Gambar 3.8 Relasi antar tabel

Dari tabel – tabel yang terbentuk selanjutnya akan dirancang struktur data dari masing – masing tabel :

1. Struktur tabel virus_komputer.

Tabel 3.1 Struktur tabel virus_komputer

Nama <i>Field</i>	Tipe	Ukuran
ID_virus	<i>Bigint</i>	10
nama_virus	<i>Varchar</i>	30
keterangan_virus	<i>Longtext</i>	-

2. Struktur tabel serang.

Tabel 3.2 Struktur tabel serang

Nama <i>Field</i>	Tipe	Ukuran
ID_virus	<i>Bigint</i>	10
ID_serang	<i>Bigint</i>	10
bulan_aktif_virus	<i>Varchar</i>	20

3. Struktur tabel sasaran_serang.

Tabel 3.3 Struktur tabel sasaran_serang

Nama <i>Field</i>	Tipe	Ukuran
ID_serang	<i>Bigint</i>	10
tempat_serang	<i>Varchar</i>	-

4. Struktur tabel memiliki_solusi.

Tabel 3.4 Struktur tabel memiliki_solusi

Nama <i>Field</i>	Tipe	Ukuran
ID_virus	<i>Bigint</i>	10
ID_solusi	<i>Bigint</i>	10

5. Struktur tabel solusi.

Tabel 3.5 Struktur tabel solusi

Nama <i>Field</i>	Tipe	Ukuran
ID_solusi	<i>Bigint</i>	10
Solusi	<i>Longtext</i>	-

6. Struktur tabel admin.

Tabel 3.6 Struktur tabel admin

Nama <i>Field</i>	Tipe	Ukuran
ID_admin	<i>Bigint</i>	10
nama_admin	<i>Varchar</i>	30
Password	<i>Varchar</i>	32

Contoh data dalam relasi tabel dapat dilihat pada Gambar 3.9 berikut ini.

Tabel virus_komputer

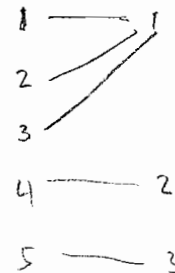
ID_virus	nama_virus	keterangan_virus
1	Macro	Menyebarkan ke semua program yang menggunakan aplikasi visual basic (<i>Ms Word, Ms Excel dan Power Point</i>).
2	Sobig.C	Memiliki kemampuan menyebar cepat melalui <i>e-mail</i> .
3	Sobig.D	Memiliki kemampuan menyebar cepat melalui Jaringan LAN.
4	ReHTML.Red	Virus ini dibuat dari program visual basic yang <i>ter-encrypt</i> .
5	March	Virus ini bekerja setiap CMOS menunjukkan tanggal 6 maret.

ID_virus	ID_serang	bulan_aktif_virus
1	1	Sewaktu - waktu
2	1	September
3	1	September
4	2	Sewaktu - waktu
5	3	Maret

ID_serang	tempat_serang
1	<i>Document</i>
2	Program (html, htm, asp, php, jsp dan vbs)
3	Boot Sector
4	File berekstensi exe

Tabel memiliki_solusi

ID_virus	ID_solusi
1	1
2	2
3	2
4	3
5	4



Tabel solusi

ID_solusi	Solusi
1	Hapus <i>File</i> MANN.DOT, USER.DOT, SNRML.DOT dan ETHAN.DOT
2	Penangkalnya bisa diakses melalui update <i>lisensi</i> antivirus panda pada masing - masing customer.
3	Men- <i>scan</i> komputer dan jika ditemukan <i>file</i> kernel32.dll, segeralah dihapus. File ini kelihatannya <i>file system</i> padahal ini adalah virus.
4	Update scanner antivirus agar virus terdeteksi.

Gambar 3.9 Contoh Data Dalam Relasi Antar Tabel

3.4.2. Perancangan Struktur XML

Perancangan struktur XML yang akan dibuat dalam sistem akan dimulai dari definisi *root*, *child* dan *subchild* dilanjutkan dengan membentuk struktur *treenya*.

3.4.2.1 Definsi *Root*, *Child* dan *Subchild*

Dalam struktur XML yang akan dibuat pada sistem tersusun atas *root*, *child* dan *subchild* dengan spesifikasi sebagai berikut :

1. *Root*.

Root node ini melingkupi keseluruhan dari dokumen XML, jadi dalam suatu dokumen XML hanya ada satu *root node*. *Root node* membentuk hirarki yang di dalamnya terdapat *child* dan *subchild*. *Root node* dalam dokumen XML pada sistem adalah ALLData.

2. *Child*.

Child node dalam dokumen XML pada sistem adalah *virus_komputer*.

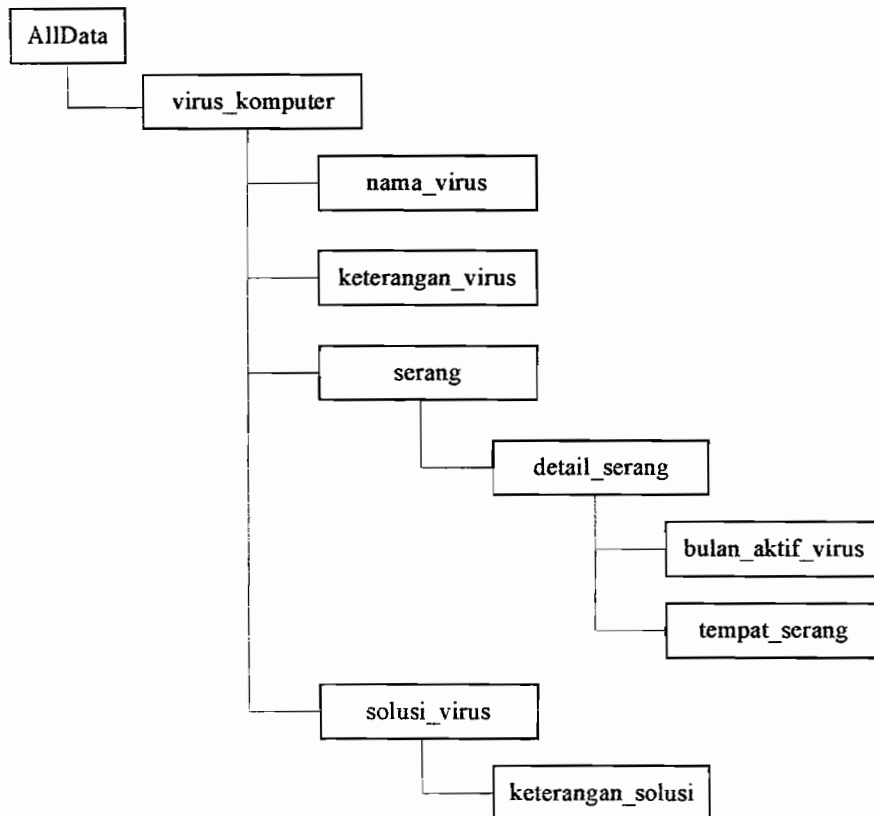
3. *Subchild*.

Subchild node dalam dokumen XML pada sistem adalah sebagai berikut :

- a. *nama_virus*.
- b. *keterangan_virus*
- c. *serang*, di dalam *serang* terdapat *subchild* lebih kecil yaitu *detail_serang* dan di dalam *detail_serang* terdapat *subchild* lebih kecil lagi yaitu *bulan_aktif_virus* dan *tempat_serang*.
- d. *solusi_virus*, di dalam *solusi_virus* terdapat *subchild* lebih kecil lagi yaitu *keterangan_solusi*.

3.4.2.2 Struktur XML

Struktur XML yang akan dibentuk dari *root node*, *child node* dan *subchild node* akan digambarkan dalam bentuk *tree* pada Gambar 3.10 berikut ini.



Gambar 3.10 Struktur XML dalam bentuk *tree*

Contoh struktur XML yang terbentuk dari *root node*, *child node* dan *subchild node* adalah sebagai berikut :

```

<?xml version="1.0" ?>
- <ALLData>
  - <virus_komputer ID_virus="2">
    <nama_virus>i love u</nama_virus>
    <keterangan_virus>virus ini ditemukan oleh Onel
      De Guszman, filipina.</keterangan_virus>
    - <serang>
      - <detail_serang>
        <bulan_aktif_virus>mei</bulan_aktif_virus>
        <tempat_serang>program (html, htm, asp,
          php, jsp dan vbs)</tempat_serang>
        </detail_serang>
      </serang>
    - <solusi_virus>
      <keterangan_solusi>hati-hati dengan
        browsing di internet.</keterangan_solusi>
      <keterangan_solusi>hati-hati dengan email
        yang mencurigakan.</keterangan_solusi>
    </solusi_virus>
  </virus_komputer>
  - <virus_komputer ID_virus="3">
    <nama_virus>melissa</nama_virus>
    <keterangan_virus>virus ini kebanyakan
      menyerang Macintosh.</keterangan_virus>
    - <serang>
      - <detail_serang>
        <bulan_aktif_virus>maret</bulan_aktif_virus>
        <tempat_serang>software
          aplikasi</tempat_serang>
        </detail_serang>
      </serang>
    - <solusi_virus>
      <keterangan_solusi>hati-hati dengan kiriman
        mail berisi "Important message
          from..."</keterangan_solusi>
    </solusi_virus>
  </virus_komputer>
</ALLData>

```

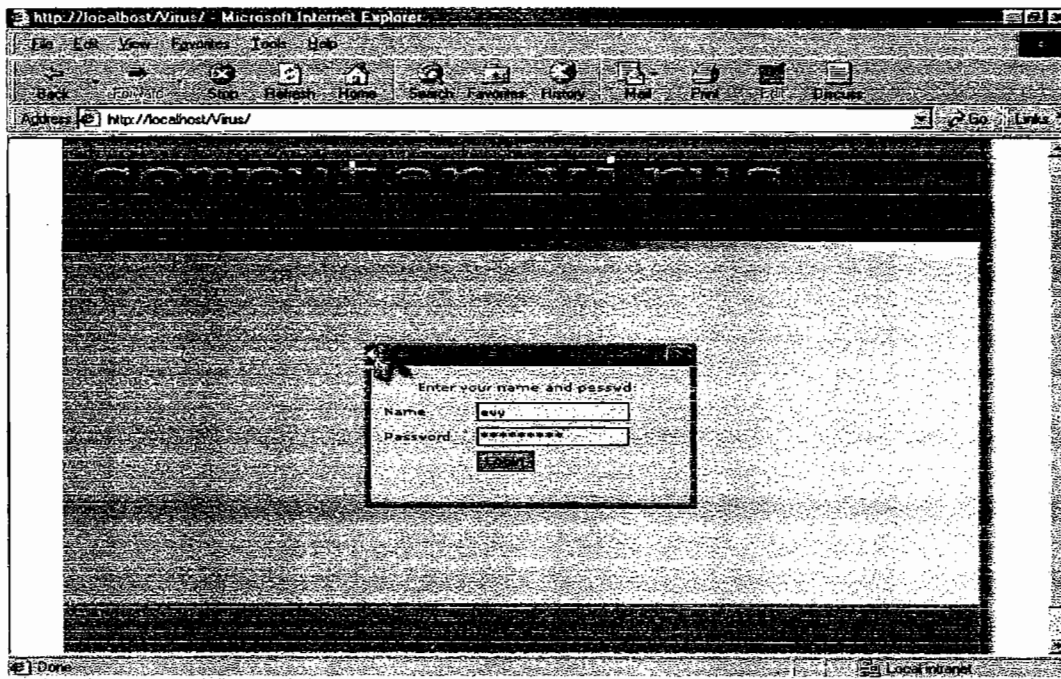
3.4.3 Perancangan Antar Muka

Pada perancangan antar muka ini terdiri dari dua antar muka, yaitu yang digunakan oleh administrator pada pihak *server* dan yang digunakan oleh *user* di pihak *client*. Perancangan dan pembahasannya adalah sebagai berikut :

1. Perancangan antar muka perangkat lunak untuk administrator.
 - a. Rancangan antar muka *login* administrator.

Rancangan antar muka *login* administrator dapat dilihat pada Gambar

3.11.

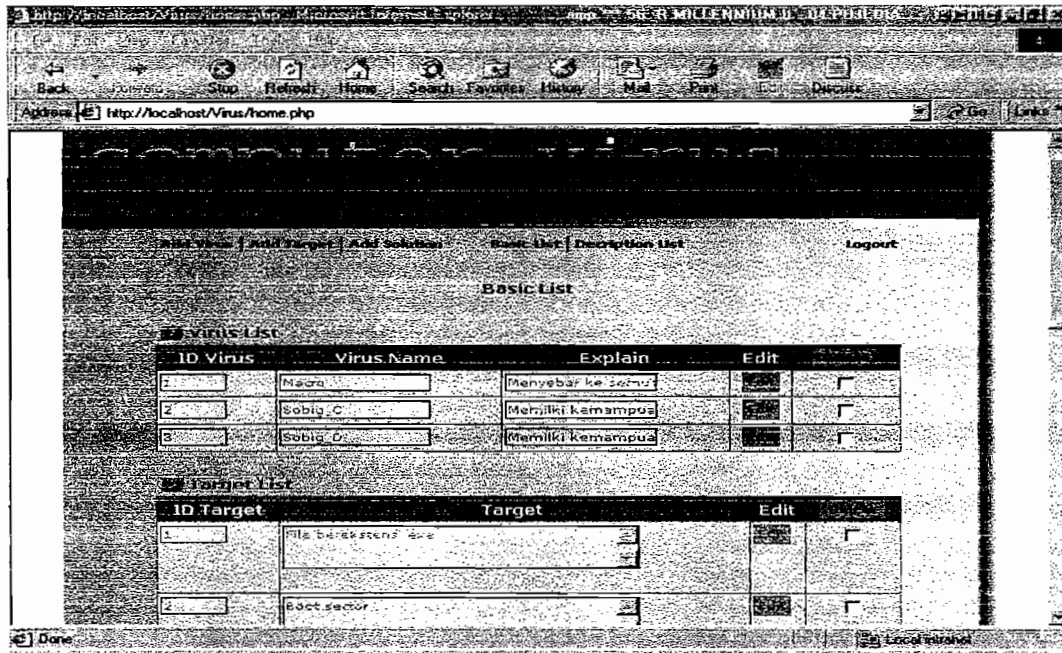


Gambar 3.11 Rancangan antar muka menu *login* administrator

Pada antar muka menu *login* ini akan tampil pada jendela *microsoft internet explorer*. Pada antar muka menu *login* ini, administrator memasukkan nama dan *password*. Jika *password* yang dimasukkan salah maka administrator akan mengulang mengisi *form password*. Apabila *password* cocok maka administrator dapat melakukan *add*, *edit* dan *delete* data – data virus komputer.

b. Rancangan antar muka *form* utama administrator.

Rancangan antar muka *form* utama administrator dapat dilihat pada Gambar 3.12.

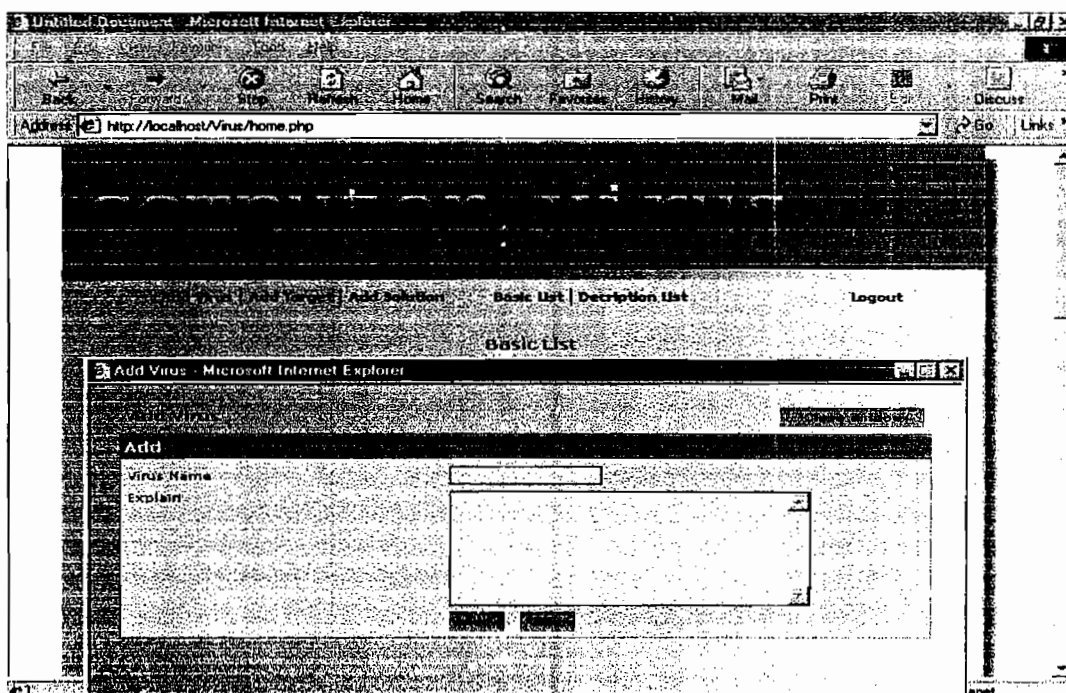


Gambar 3.12 Rancangan antar muka menu utama administrator

Pada rancangan antar muka menu utama ini, terdapat beberapa menu yaitu *add virus*, *add target*, *add solution*, *basic list*, *description list* dan *logout*. Pada halaman menu utama ini akan ditampilkan tampilan *basic list* yang dilengkapi edit dan *delete per - record*.

c. Rancangan antar muka tambah data virus komputer.

Rancangan antar muka tambah data virus komputer dapat dilihat pada Gambar 3.13.

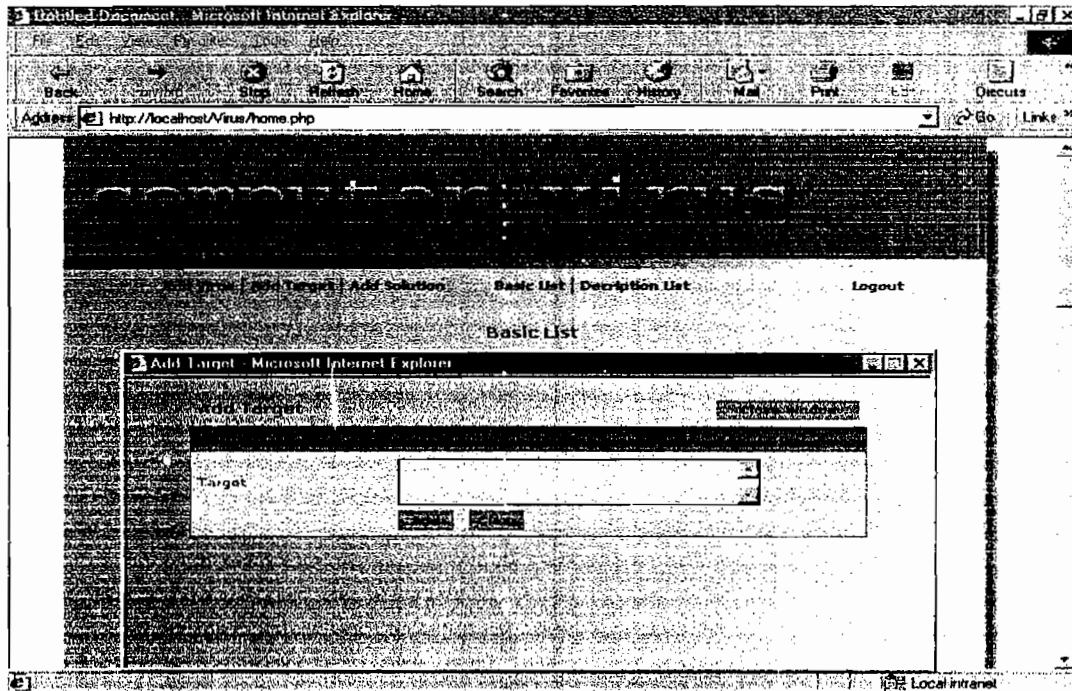


Gambar 3.13 Rancangan antar muka *form* tambah data virus komputer

Pada rancangan antar muka *form* tambah data virus komputer ini, administrator dapat melakukan tambah data dengan memasukkan *virus name*, *date* dan *explain*. Tombol *add* digunakan untuk menambahkan data virus ke dalam *database* dan tombol *clear* digunakan untuk membersihkan teks jika administrator menginginkan untuk menambah data secara berulang. *Close window* digunakan untuk keluar dari jendela tambah data ini.

d. Rancangan antar muka tambah data tempat sasaran serang virus komputer.

Rancangan antar muka tambah data tempat sasaran serang virus komputer dapat dilihat pada Gambar 3.14.



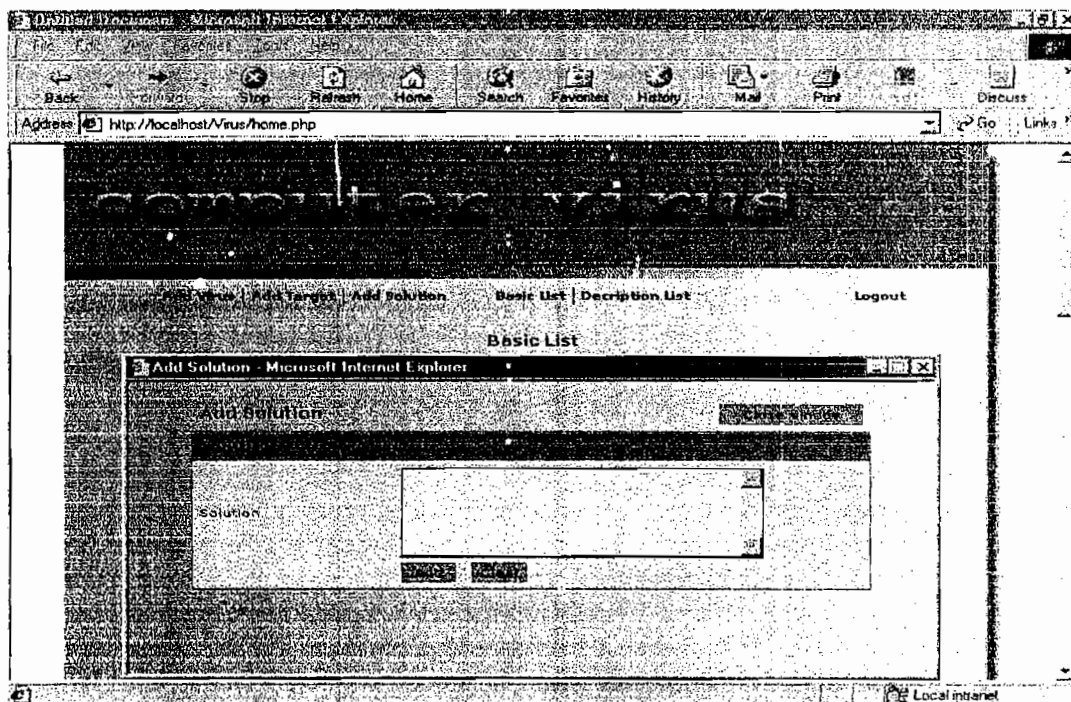
Gambar 3.14 Rancangan antar muka tambah data tempat sasaran virus komputer

Pada rancangan antar muka *form* tambah data tempat sasaran virus komputer ini, administrator dapat melakukan tambah tempat sasaran virus komputer. Pada *form* tersebut administrator akan menuliskan *target* sasaran virus komputer di sebuah tempat atau berbagai tempat. Tombol *add* digunakan untuk menambahkan data *target* ke dalam *database* dan tombol *clear* digunakan untuk membersihkan teks jika administrator menginginkan untuk menambah data secara berulang. *Close window* digunakan untuk keluar dari jendela tambah data tempat sasaran virus ini.



e. Rancangan antar muka tambah data solusi.

Rancangan antar muka tambah data solusi dapat dilihat pada Gambar 3.15.

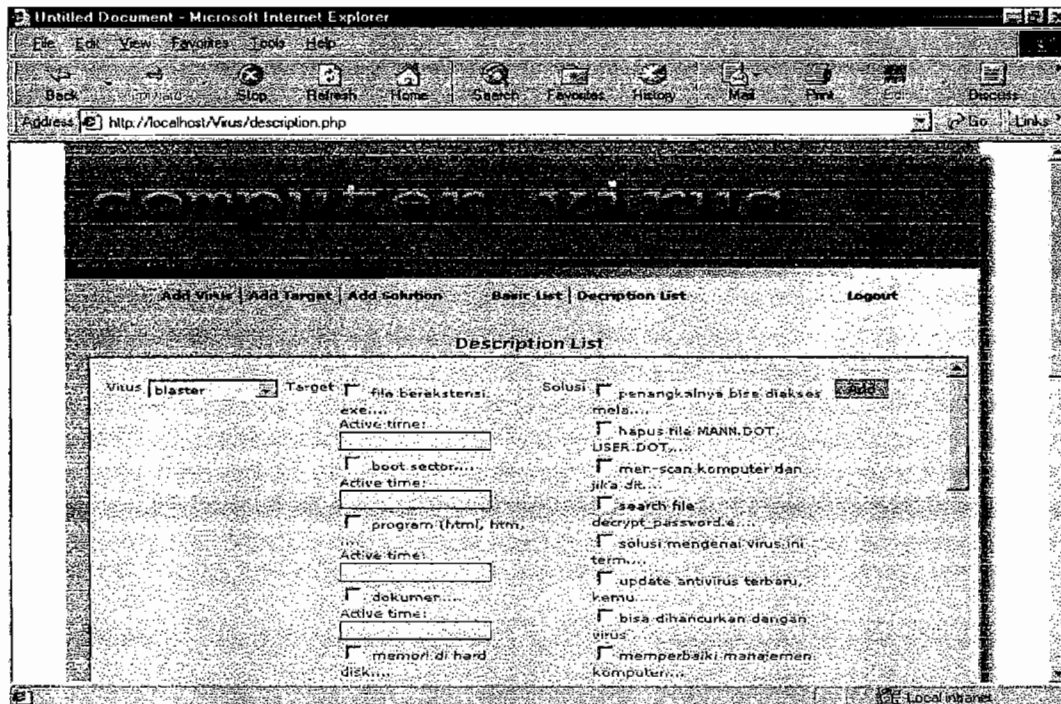


Gambar 3.15 Rancangan antar muka tambah data solusi

Pada rancangan antar muka *form* tambah data solusi ini, administrator dapat melakukan tambah data solusi. Pada *form* tersebut administrator akan menuliskan solusi mengenai penyerangan virus, baik satu atau lebih solusi. Tombol *add* digunakan untuk menambahkan data solusi ke dalam *database* dan tombol *clear* digunakan untuk membersihkan teks jika administrator menginginkan untuk menambah data secara berulang. *Close window* digunakan untuk keluar dari jendela tambah data solusi ini.

f. Rancangan antar muka *Description list*.

Rancangan antar muka *Description list* dapat dilihat pada Gambar 3.16.

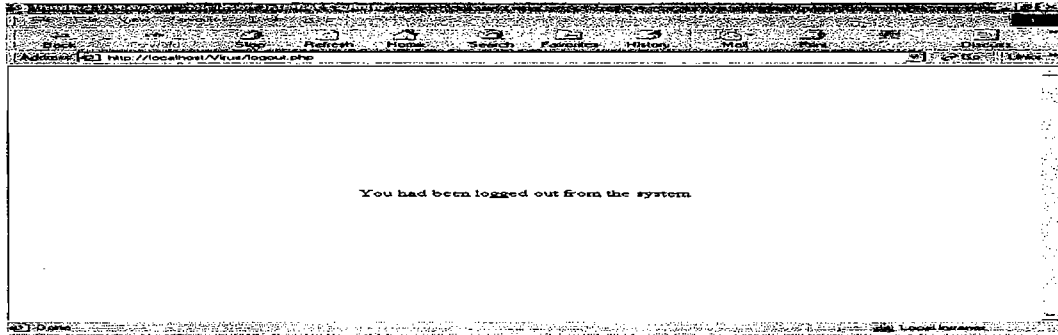


Gambar 3.16 Rancangan antar muka *description list*

Pada antar muka *description list* ini berisi tentang data – data virus komputer yang telah dimasukkan oleh administrator. Jadi data – data yang ada dalam *form* ini telah ter - *update* oleh administrator. Pada saat administrator menambah data virus komputer, selanjutnya administrator akan masuk pada *form description list* ini karena data virus komputer yang ditambahkan akan masuk dalam daftar *check box* yang ada dalam *form description list* kemudian administrator memilih data yang ingin ditambah dengan cara menekan tombol *add*.

g. Rancangan antar muka untuk *logout* program administrator.

Rancangan antar muka untuk *logout* program administrator dapat dilihat pada Gambar 3.17.

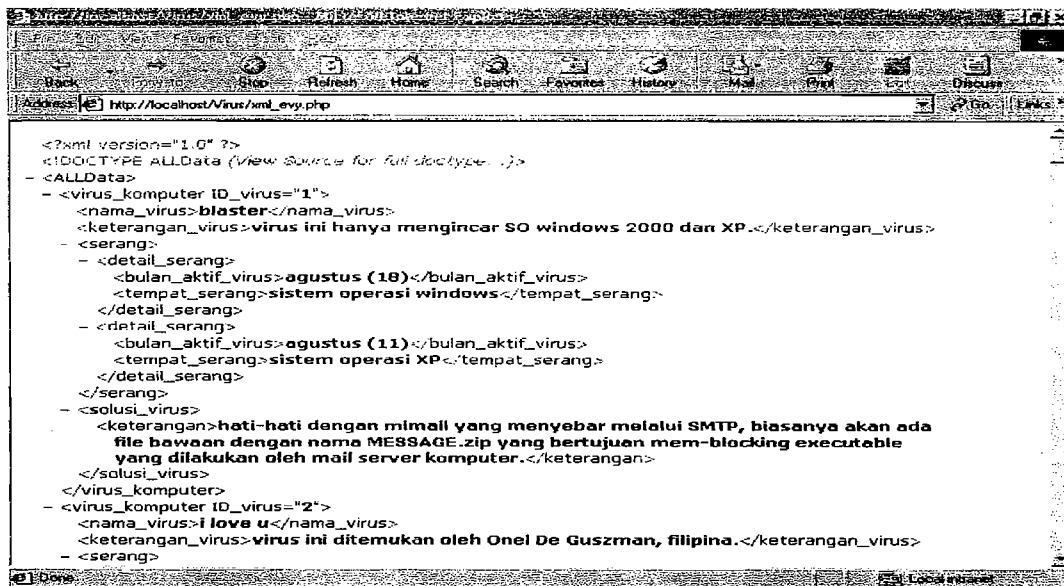


Gambar 3.17 Rancangan antar muka *logout* program

Pada bagian ini hanyalah sebuah keterangan yang menyatakan bahwa administrator melakukan *logout* program.

h. Rancangan halaman XML pada *internet explorer*.

Rancangan halaman XML pada *internet explorer* dapat dilihat pada Gambar 3.18.



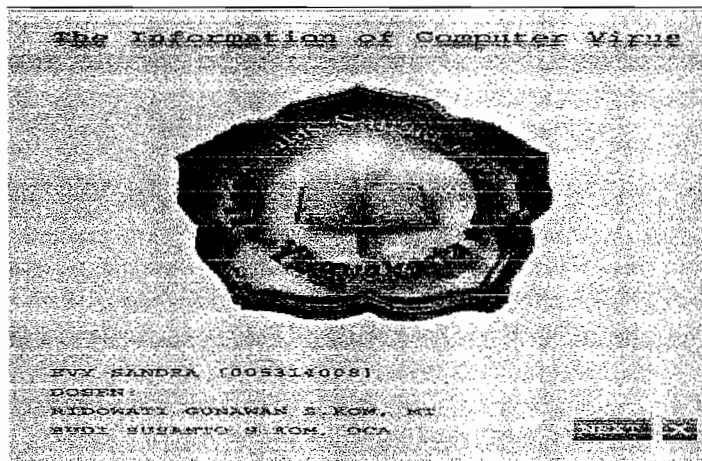
Gambar 3.18 Rancangan halaman XML pada *internet explorer*

Pada halaman XML ini dibuat dengan skrip PHP (xml_ovy.php), semua data virus dalam *database* virus akan diubah dalam format XML sehingga data virus yang diinginkan *user* bisa diakses pada aplikasi yang berbeda, seperti Visual Basic. Jadi pada saat pihak *client* (Visual Basic) mengakses data virus, data yang diakses bukan lagi data dalam bentuk *database* tetapi dalam format XML.

2. Rancangan antar muka untuk *user* pihak *client*.

a. Rancangan antar muka untuk tampilan awal.

Rancangan antar muka untuk tampilan awal nampak pada Gambar 3.19.

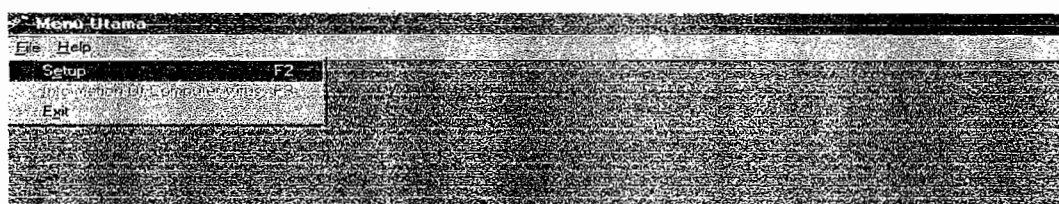


Gambar 3.19 Perancangan tampilan awal

Pada *form* tampilan awal ini, *next* berarti masuk ke program dan tanda *x* (*exit*) berarti keluar program.

b. Rancangan antar muka menu utama

Rancangan antar muka menu utama dapat dilihat pada Gambar 3.20.

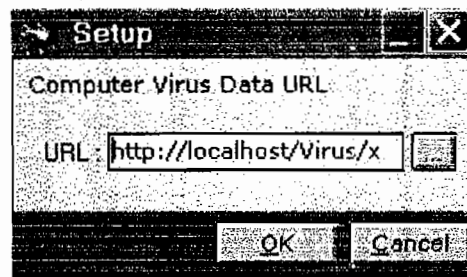


Gambar 3.20 Rancangan antar muka menu utama

Pada menu ini terdapat dua menu yaitu menu *file* dan menu *help*. Menu *file* berisi submenu *setup*, *information of virus computer* dan *exit*, Menu *help* berisi submenu *about programmer* dan *about program*.

c. Rancangan antar muka *setup*.

Rancangan antar muka *setup* dapat dilihat pada Gambar 3.21.

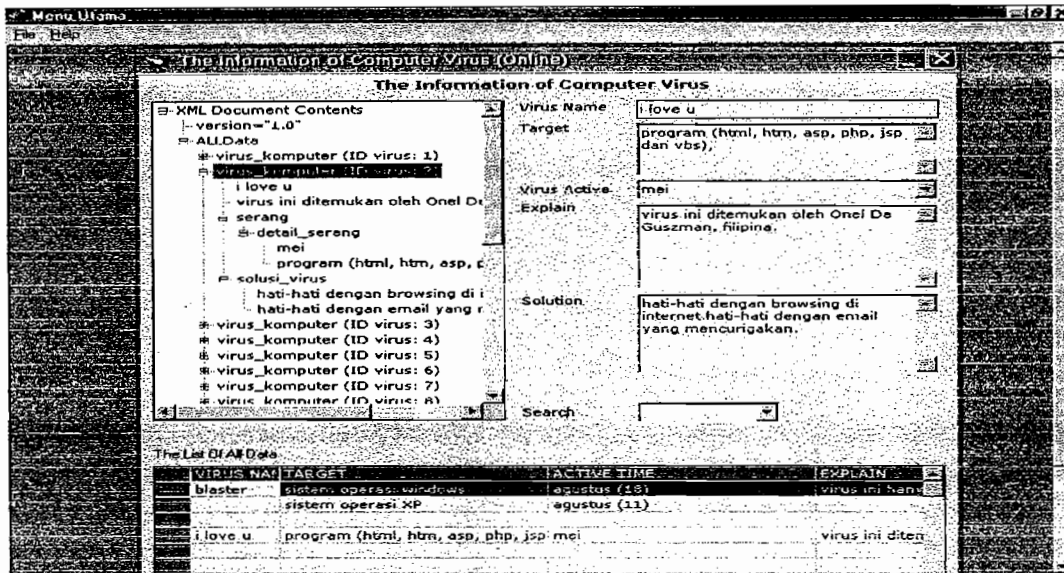


Gambar 3.21 Rancangan antar muka untuk *setup*

Pada rancangan antar muka *setup* ini digunakan untuk meletakkan alamat URL (*Uniform Resource Locator*) tempat data – data virus komputer disimpan, jadi jika alamat URL telah ganti maka langsung saja mengganti pada *form* ini. Jika URL yang diisikan lewat *text box http* berarti data XML yang di ambil dari alamatnya bersifat *on line* dan apabila URL di ambil dari *command dialogue* berarti data XML yang di ambil bersifat *off line* (lokal).

d. Rancangan antar muka untuk informasi virus komputer.

Rancangan antar muka untuk informasi virus komputer dapat dilihat pada Gambar 3.22.



Gambar 3.22 Rancangan antar muka untuk informasi virus komputer

Pada *form* informasi virus komputer ini akan menampilkan data – data virus komputer dalam bentuk *treeview*, *text box* dan *msflexgrid*. *Search* digunakan untuk mencari data yang diinginkan. *Search* data dapat dicari berdasarkan *virus_name*, *active_time* dan *target* virus komputer.

e. Rancangan antar muka untuk pencarian data berdasarkan nama virus.

Rancangan antar muka untuk pencarian data berdasarkan nama virus dapat dilihat pada Gambar 3.23.

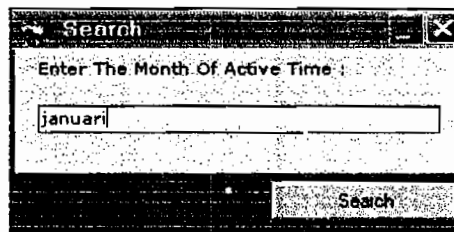


Gambar 3.23 Rancangan antar muka untuk cari data berdasarkan nama virus

Pada menu *search* ini, *user* pada pihak *client* dapat melakukan pencarian data untuk mendapatkan data yang diinginkan. *User* akan memasukkan nama virus komputer yang akan dicari dan jika data yang dicari ada maka akan ditampilkan di layar monitor, tetapi jika data yang dicari tidak ada maka data tidak ditampilkan.

f. Rancangan antar muka untuk pencarian berdasarkan data bulan aktif virus.

Rancangan antar muka untuk pencarian berdasarkan data bulan aktif virus dapat dilihat pada Gambar 3.24.

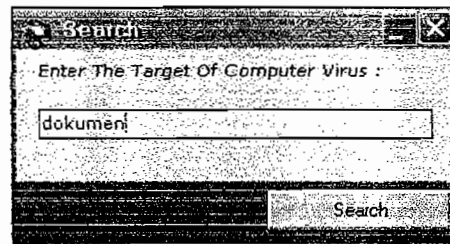


Gambar 3.24 Rancangan antar muka untuk cari berdasarkan bulan aktif virus

Pada menu *search* ini, *user* pada pihak *client* dapat melakukan pencarian data untuk mendapatkan data yang diinginkan. *User* akan memasukkan bulan aktif virus komputer yang akan dicari dan jika data yang dicari ada maka akan ditampilkan di layar monitor, tetapi jika data yang dicari tidak ada maka data tidak ditampilkan.

g. Rancangan antar muka untuk pencarian data berdasarkan tempat serang virus.

Rancangan antar muka untuk pencarian data berdasarkan tempat serang virus dapat dilihat pada Gambar 3.25.

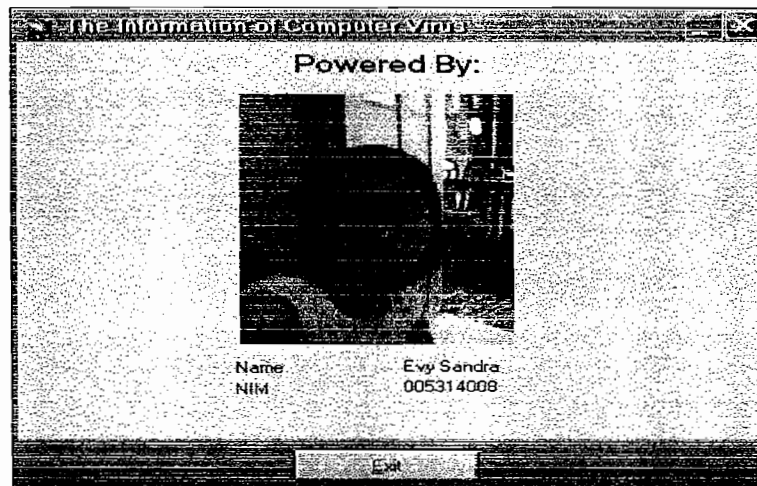


Gambar 3.25 Rancangan antar muka untuk cari berdasarkan target serang virus

Pada menu *search* ini, *user* pada pihak *client* dapat melakukan pencarian data untuk mendapatkan data yang diinginkan. *User* akan memasukkan target serang virus komputer yang akan dicari dan jika data yang dicari ada maka akan ditampilkan di layar monitor, tetapi jika data tidak ada maka data tidak akan ditampilkan.

h. Rancangan antar muka menu *help* untuk *about programmer*.

Rancangan antar muka menu *help* untuk *about programmer* dapat dilihat pada Gambar 3.26.

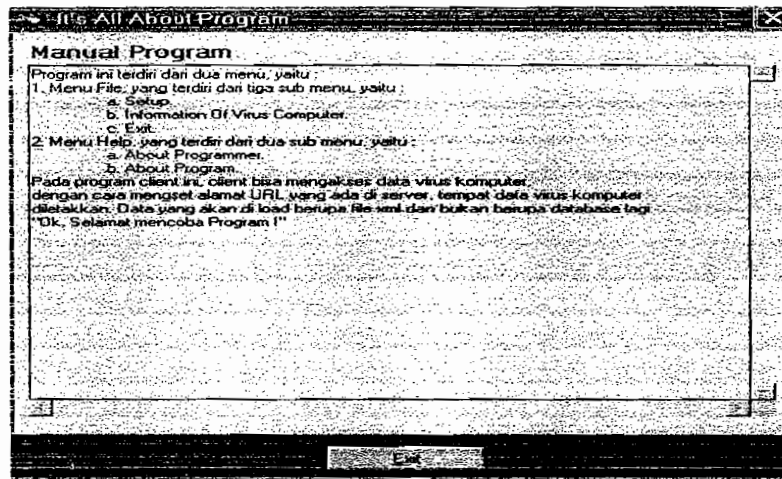


Gambar 3.26 Rancangan antar muka *form about programmer*

Pada *form about programmer* ini, berisi mengenai foto dan identitas pembuat program. Tombol *exit* digunakan untuk keluar dari *form about programmer*.

i. Rancangan antar muka menu *help* untuk *form about program*.

Rancangan antar muka menu *help* untuk *form about program* dapat dilihat pada Gambar 3.27.



Gambar 3.27 Rancangan antar muka *form about program*

Pada *form about program* ini, berisi mengenai langkah – langkah jalannya program. Tombol *exit* digunakan untuk keluar dari *form about program*.

3.5 Analisis Sumber Daya

Analisis sumber daya yang dilakukan merupakan analisis sumber daya yang akan dipakai pada saat uji coba program.

3.5.1 Sumber Daya Manusia (*Brainware*)

1. *Server Side*.

- a. Minimal dapat mengoperasikan komputer dengan SO *Windows*.
- b. Menguasai proses tambah, *edit* dan hapus data.
- c. Menguasai PHP, *database* MySQL, XML dan Visual Basic untuk mengembangkan sistem.

- d. Mengetahui perkembangan dunia virus komputer sehingga memudahkan dalam melakukan *update* data – data virus komputer.
2. *Client*.
 - a. Memahami cara pengoperasian dengan aplikasi Visual Basic.
 - b. Memahami konsep *load* data dari *server* ke *client*.
 - c. Dapat melakukan proses *searching* data – data virus komputer.

3.5.2 Perangkat Lunak (*Software*)

1. *Server Side*.
 - a. Sistem Operasi *Windows ME (Millinneum Edition)*.
 - b. *Dreamweaver MX*.
 - c. *Browser Internet Explorer* minimum versi 5.5.
 - d. PHP 4.0.5.
 - e. *Database MySQL 3.23.32*.
 - f. *Server Web Apache 1.3.14*.
 - g. *Adobe Photoshop 7.0*.
2. *Client Side*.
 - a. Sistem Operasi *Windows ME (Millinneum Edition)*.
 - b. *Visual Basic 6.0*.
 - c. *Adobe Photoshop 7.0*.

3.5.3 Perangkat Keras (*Hardware*)

Perangkat Keras untuk *server side* dan *client side* dalam uji coba program adalah sebagai berikut :

1. *Server Side.*

- a. Processor IP-III.
- b. Memory 64 MB SDRAM.
- c. HDD 40 GB 7200 RPM.
- d. VGA Card 64MB.
- e. Sound Card.
- f. LAN Card.

2. *Client Side.*

- a. Processor IP-III.
- b. Memory 64 MB SDRAM.
- c. HDD 40 GB 7200 RPM.
- d. VGA Card 64MB.
- e. Sound Card.
- f. LAN Card

BAB 4

IMPLEMENTASI SISTEM

Pada sistem aplikasi XML dalam pengaksesan *database server* ini dapat dilakukan oleh setiap *user* yang menginginkan untuk dapat mengolah sendiri terhadap data yang diperoleh. Artinya, bahwa setiap *user* dapat melakukan sistem ini jika *user* menginginkan suatu model pertukaran data yang bersifat umum. Maka dari itu dengan sistem ini mengimplementasikan dokumen XML untuk pertukaran data antar *database server* (MySQL) dengan aplikasi yang berbeda (Visual Basic) melalui *web server* dan PHP sehingga diperoleh data yang bersifat umum. Pada implementasi ini digunakan studi kasus *database* virus komputer, jadi sistem ini sekaligus memberikan manfaat bagi *user* pihak *client* untuk mengakses informasi data – data virus komputer yang telah diisikan oleh administrator di pihak *server*.

Sistem ini terdiri dari pihak *server* dan *client*. Pihak *server* merupakan pihak yang dapat mengakses, menambah, mengedit dan menghapus data – data virus komputer dalam *database server*. Selain itu administrator juga bertugas mengubah data – data virus komputer ke dalam bentuk format XML dengan bentuk *file* berekstensi php. Pihak *client* merupakan pihak yang akan mengakses data – data virus komputer dalam format umum dan bukan dalam bentuk *database* lagi. Pihak *client* ini dapat melakukan operasi *search* untuk mencari data – data virus komputer berdasarkan *name_virus*, *active_time* dan *target* virus komputer sehingga diperoleh data virus komputer sesuai keinginan *user*.

Sistem ini adalah suatu sistem yang dikembangkan agar *user* mendapat terobosan baru untuk melakukan pertukaran data dalam bentuk umum, yang sekaligus berfungsi untuk menampilkan informasi data – data virus komputer pada pihak *client* yang dapat secara dipanggil *browser internet explorer* dan aplikasi Visual Basic.

4.1 Implementasi Program

Seperti dijelaskan pada bab sebelumnya, dalam aplikasi XML dalam pengaksesan *database server* ini melibatkan bahasa pemrograman PHP, XML dan Visual Basic. Di sini akan dijelaskan bagaimana program dibentuk dan dibuat, berdasarkan masing – masing tampilan atau *form* yang ada.

4.1.1 Koneksi Database

Pertama kali yang harus dilakukan dalam aplikasi XML ini adalah menghubungkan *database* dengan sistem pada pihak *server*. Pada pihak *server* akan dilakukan koneksi database *server* MySQL dengan PHP. Sintaks yang digunakan untuk mengkoneksikan MySQL dengan PHP adalah sebagai berikut :

```
<?
    mysql_connect ( " $dbhost", "$dbuser ", "$pswd ");
    mysql_select_ db ( " $namadb ");
?>
```

Variabel ‘ \$dbhost’, ‘ \$dbuser ‘, ‘ \$pswd ‘ adalah nama *host*, nama *user* dan nama *password server* yang akan dihubungkan. Sintaks yang digunakan untuk mengkoneksikan *database server* MySQL dengan PHP dalam pembuatan program ini adalah sebagai berikut :

```
<?
    $host=mysql_connect("localhost", "evy", "110382") or
    die(mysql_error());
    mysql_select_db("virus", $host) or die(mysql_error());
?>
```

Dalam sintaks tersebut nama *host* adalah *localhost*, nama *user* adalah *evy* dan *passwordnya* 110382. Koneksi *database* ini akan mengkoneksikan *database* *virus* dengan PHP dan koneksi *database* ini akan secara otomatis terputus pada saat skrip program selesai dijalankan. Untuk memudahkan koneksi maka sintaks ini akan disimpan dalam *file connect.php* yang berada dalam sebuah *folder connection* tersendiri. Jika sistem akan melakukan koneksi dengan *database* maka *file* yang berisi skrip koneksi *database* ini akan dipanggil.

4.1.2 Modul Antarmuka Pihak Server

4.1.2.1 Modul *Index Administrator*

Modul ini merupakan halaman utama bagi administrator (*index.php*). Modul ini akan meminta *input* berupa *name* dan *password* administrator, *input* dari administrator akan diproses untuk menentukan apakah orang yang memasukkan data berhak mengakses halaman berikutnya. Inti program untuk *form* ini adalah sebagai berikut :

```

<?
session_start();
include "connection/connect.php";
include "includes/checkfirst.php";
include "includes/fungsi.php";
print_header();
?>
<table align="center">
  <tr>
    <td background="images/formlogin_18.gif" width="250"
      height="149" align="center">
      <table width="90%" class="tekshitam">
        <tr>
          <td align="center">
            Enter your name and passwd:
          </td>
        </tr>
      </table>
      <table width="90%" class="tekshitam">
        <form action="" method="post">
          <tr>
            <td>
              Name
            </td>
            <td>
              <input type="text" name="nama" class="textbox">
            </td>
          </tr>
          <tr>
            <td>
              Password
            </td>
            <td>
              <input type="password" name="password"
                class="textbox">
            </td>
          </tr>
          <tr>
            <td>
            </td>
            <td>
              <input type="submit" name="login"
                value="Login" class="button">
            </td>
          </tr>
        </form>
      </table>
    </td>
  </tr>
</table>
<?
print_footer();
?>

```


Untuk melakukan proses otentikasi administrator modul *index* akan mengirimkan data *name* dan *password* setelah menekan tombol *login* ke modul *checkfirst.php*. Dalam kode program *form index* menyertakan modul *checkfirst.php*. Setelah menyertakan modul *checkfirst.php* maka akan dicek apakah sistem telah membuat *file session* jika sudah maka *file* ini akan disertakan. Data yang diterima modul *checkfirst.php* akan dicocokkan dengan data yang ada di dalam modul *index.php*. Kode program untuk pengecekan data yang dimasukkan administrator dengan data yang ada dalam *database* adalah sebagai berikut :

```
if(isset($_HTTP_POST_VARS["login"])){
    $result=mysql_query("select 1 from admin
        where nama_admin='".$_$_HTTP_POST_VARS["nama"]."' and
        password='".$_$_HTTP_POST_VARS["password"]."'")or
    die(mysql_error());
    $row=mysql_fetch_row($result);
```

Jika data yang dimasukkan benar maka sistem akan membuat sebuah *file session* yang berisi *name* dan *password*. *Session* ini digunakan untuk proses otentikasi selanjutnya tanpa harus memasukkan data lagi. *Session* ini dibuat setiap kali administrator melakukan *login*. Kode program untuk membuat *session* dan mengisi nilai *session* adalah sebagai berikut :

```
if($row[0]==1){
    session_register("nama");
    session_register("password");
    $_HTTP_SESSION_VARS["nama"]=$_HTTP_POST_VARS["nama"]
    $_HTTP_SESSION_VARS["password"]=$_HTTP_POST_VARS["password"];
}
```

Jika program sudah melakukan proses *session* maka program akan berpindah ke halaman menu utama administrator (*home.php*) dengan menggunakan kode program sebagai berikut :

```
echo"<metahttpequiv=\"refresh\"content=\"0;URL=home.php\">";
```

Jika pengecekan data gagal, dalam artian tidak sama dengan 1 atau data yang dimasukkan administrator tidak sesuai dengan data yang ada dalam *database* maka administrator tidak bisa masuk ke halaman selanjutnya dan sistem akan kembali ke *form index*. Pada modul *index* ini administrator akan mengulang memasukkan data hingga data tersebut cocok dengan data yang ada dalam *database*.

Pada *modul index* selain menyertakan modul *checkfirst.php* juga menyertakan modul *fungsi.php*. Dalam modul *index* ini, modul *fungsi.php* digunakan untuk menampung fungsi *print_header* dan fungsi *print_footer* yang digunakan untuk mempercantik tampilan.

4.1.2.2 Modul utama administrator

Pada menu administrator ini akan ditampilkan jika *input* nama administrator dan *password* sesuai dengan data yang ada dalam *database* admin. Pada menu utama administrator ini merupakan menu *basic list* yang disimpan dalam *home.php*. Pada menu utama administrator ini menyajikan 6 buah *link*, yaitu :

1. *Add Virus*, yang berfungsi bagi administrator untuk menambah virus beserta *explain* virus yang ditambahkan sehingga pada tampilan *virus list* akan bertambah virus sebanyak virus yang administrator masukkan.

2. *Add Target*, yang berfungsi bagi administrator untuk menambah *target* serang dari virus sehingga akan menambah data *target* pada *target list*.
3. *Add Solution*, yang berfungsi bagi administrator untuk menambah solusi memberantas virus sehingga akan menambah data solusi pada *solution list*.
4. *Basic List*, yang berfungsi bagi administrator untuk melihat daftar *virus list*, *target list* dan *solution list*. Pada menu *basic list* sekaligus merupakan menu utama administrator, jadi administrator akan melakukan pemilihan menu pada *form* ini.
5. *Description List*, yang berfungsi bagi administrator untuk melakukan proses *add* secara lengkap dari data yang telah dimasukkan sebelumnya oleh administrator pada *form basic list*.

Pada *form home* ini akan menampilkan *virus list*, *target list* dan *solution list* yang telah dilengkapi dengan fasilitas *edit* dan *delete*. Kode program yang digunakan untuk *update virus list* dengan fasilitas *edit* dan *delete* adalah sebagai berikut :

```

$resultjml=mysql_query("select          max(ID_virus)          from
virus_komputer");
$rowjml=mysql_fetch_row($resultjml);

init_print_nav(5);

for($i=1;$i<=$rowjml[0];$i++){
    if(isset($_HTTP_POST_VARS["save1".$i])){
        mysql_query("update virus_komputer
set nama_virus='".$_HTTP_POST_VARS["namavirus".$i]."',
keterangan_virus='".$_HTTP_POST_VARS["keteranganvirus".$i
]."'
where
ID_virus='".$_HTTP_POST_VARS["tmpidvirus".$i]."'") or
die(mysql_error());
    }
}

```

```

if(isset($_HTTP_POST_VARS["delete1"])){
    if($_HTTP_POST_VARS["delete"]){
        while(list($key,$value)=each($_HTTP_POST_VARS["delete"])){
            mysql_query("delete from virus_komputer
                where ID_virus='".$value."'");
        }
    }
}

```

Kode program yang digunakan untuk *update target list* dengan fasilitas *edit* dan

delete adalah sebagai berikut :

```

$resultjml=mysql_query("select          max(ID_serang)          from
sasaran_serang");
$rowjml=mysql_fetch_row($resultjml);

for($i=1;$i<=$rowjml[0];$i++){
    if(isset($_HTTP_POST_VARS["save2".$i])){
        mysql_query("update sasaran_serang
            set tempat_serang='".$_HTTP_POST_VARS["target".$i]."'
            where
            ID_serang='".$_HTTP_POST_VARS["tmpidtarget".$i]."'")    or
            die(mysql_error());
        }
    }

    if(isset($_HTTP_POST_VARS["delete2"])){
        if($_HTTP_POST_VARS["delete"]){
            while(list($key,$value)=each($_HTTP_POST_VARS["delete"]))
            {
                mysql_query("delete from sasaran_serang
                    where ID_serang='".$value."'");
            }
        }
    }
}

```

Kode program yang digunakan untuk *update solution list* dengan fasilitas *edit* dan

delete adalah sebagai berikut :

```

$resultjml=mysql_query("select max(ID_solusi) from solusi");
$rowjml=mysql_fetch_row($resultjml);

for($i=1;$i<=$rowjml[0];$i++){
    if(isset($_HTTP_POST_VARS["save3".$i])){
        mysql_query("update solusi
            set solusi='".$_HTTP_POST_VARS["solusi".$i]."'
            where
            ID_solusi='".$_HTTP_POST_VARS["tmpidsolusi".$i]."'")    or
            die(mysql_error());
        }
    }
}

```

```

if(isset($_HTTP_POST_VARS["delete3"])){
    if($_HTTP_POST_VARS["delete"]){
        while(list($key,$value)=each($_HTTP_POST_VARS["delete"]){
            mysql_query("delete from solusi
                where ID_solusi='".$value."'");
        }
    }
}

```

4.1.2.3 Modul Add Virus

Modul *add virus* (*addvirus.php*) ini digunakan untuk membuat *form add virus* yang digunakan oleh administrator untuk menambah data virus komputer terbaru. Pada menu *add virus* ini terdiri dari dua *field*, yaitu *virus name* dan *explain*. Pada menu *add virus* ini terdapat tombol *add* untuk menambahkan data yang sudah dimasukkan ke dalam *field add virus* dan tombol *clear* digunakan untuk membersihkan *field* dari teks yang telah terisi data. Tombol *close window* digunakan untuk keluar dari *form* ini dan kembali ke menu utama administrator (*home.php*).

Pada *form add virus* ini pada saat selesai mengisi data *virus name* dan *explain* kemudian menekan tombol *add* maka secara otomatis sistem akan menjalankan *insert* data ke dalam *database* *virus_komputer*. Kode program *insert* data tersebut adalah sebagai berikut :

```

if(isset($_HTTP_POST_VARS["add"])){
    if(!empty($_HTTP_POST_VARS["namavirus"])){
        mysql_query("insert into virus_komputer
            values('','".$_HTTP_POST_VARS["namavirus"]."',
                '".$_HTTP_POST_VARS["keteranganvirus"]."')");
    }
    else{
        $error="Error..... Virus Name must be fill...";
    }
}

```

Pada kode program *insert* data ini, jika administrator menekan tombol *add* dan *field* nama virus tidak kosong atau telah ada pengisian data dalam *field* tersebut maka akan sistem akan melakukan *insert* data. Kode program yang digunakan untuk melakukan *insert* data adalah sebagai berikut :

```
mysql_query("insert into virus_komputer
values('','$HTTP_POST_VARS["namavirus"]."',
'".$HTTP_POST_VARS["keteranganvirus"]."');
");
```

Pada *form add virus* ini terdapat tombol *close window* yang digunakan untuk keluar dari *form add virus*. Sintaks yang digunakan dalam *close windows* ini adalah sintaks *java script*, sintaksnya adalah sebagai berikut :

```
<script language="JavaScript1.2">
function closewindow(){
window.close();
}
</script>
```

4.1.2.4 Modul *Add Target*

Pada modul *add target* ini (*addtarget.php*) digunakan untuk membuat *form add target* yang digunakan oleh administrator untuk menambah data *target* virus komputer yang terbaru. Pada *form add target* ini terdapat satu *field*, yaitu *target*. Pada *form add target* ini, mengenai bentuk tampilan dan kode program untuk *insert* data prinsipnya sama dengan modul *add virus*.

4.1.2.5 *Form Add Solution*

Pada modul *add solution* ini (*addsolution.php*) digunakan untuk membuat *form add solution* yang digunakan oleh administrator untuk menambah data solusi virus komputer yang terbaru. Pada *form add solution* ini terdapat satu *field*, yaitu *solution*. Pada *form add solution* ini, mengenai bentuk tampilan dan kode program untuk *insert* data prinsipnya sama dengan *form add virus*.

4.1.2.6 Modul *Description List*

Pada modul *description list* ini (*description.php*) digunakan untuk membuat *form description list*, yang digunakan oleh administrator untuk melakukan proses tambah data – data virus komputer secara lengkap dari data yang telah diisikan sebelumnya, seperti *virus name*, *target* dan *solution*. Jadi pada *form* ini administrator akan memilih *virus name* yang akan ditambahkan, *target* yang akan ditambahkan dan *solution* yang akan ditambahkan. Pada *form description list* terdapat *text box active time* yang akan diisi oleh administrator mengenai waktu aktif virus komputer. Setelah administrator mengisi *text box active time* dan memilih *virus name*, *target* dan *solution* maka administrator akan menekan tombol *add* yang secara otomatis data virus komputer tersebut akan tersimpan dalam suatu tampilan tabel dalam *form* ini. Pada *form* ini terdapat fasilitas *edit* pada tabel yang berisi data – data virus komputer, pada saat administrator ingin mengedit data virus maka administrator akan mengklik tombol *edit* pada nama virus yang akan diedit. Pada saat *edit* diklik maka tombol *edit* akan aktif dan langsung dapat dilakukan perubahan terhadap data yang akan diedit.

Pada modul *description list*, terdapat sebuah *frame* yang digunakan untuk melakukan penambahan *virus name*, *target*, *active time* dan *solution*. *Frame* ini dibuat dari sebuah halaman *target_solusi.php*. Sintaks dalam *description list* yang digunakan untuk memanggil *frame target_solusi* adalah sebagai berikut :

```
<iframe src="target_solusi.php?id=<?=$id?>" width="95%"
height="400" frameborder="1" scrolling="auto" ></iframe>
```

Kode program yang digunakan untuk menampilkan virus komputer dengan model *text box selected* adalah sebagai berikut :

```

<select name="virus" class="textbox">
  <?
  while($rowvirus=mysql_fetch_row($resultvirus)){
    ?>
      <option value="<?=$rowvirus[0]?>"
    <?
      if($HTTP_GET_VARS["id"]== $rowvirus[0]){
        echo " selected ";
      }
    ?><?=$rowvirus[1]?></option>
  <?
  }
  ?>
</select>

```

Kode program yang digunakan untuk menampilkan data – data target dalam sebuah *frame* adalah sintaks *query database*, yaitu :

```

<?
$i=1;
while($rowtarget=mysql_fetch_row($resulttarget)){
  $resultedit=mysql_query("select serang.ID_serang
  from virus_komputer,serang
  where
  serang.ID_virus=virus_komputer.ID_virus and
  virus_komputer.ID_virus='".$HTTP_GET_VARS["id"]."') or
  die(mysql_error());
  ?>

```

Sedangkan kode program yang digunakan untuk menampilkan *text box active_time* adalah sebagai berikut :

```

<input name="masaaktif<?=$rowtarget[0]?>" value="<?=$row[2]?>"
class="textbox">

```

Kode program yang digunakan untuk menampilkan data – data solusi adalah sebagai berikut :

```

<?
  while($rowsolusi=mysql_fetch_row($resultsolusi)){
    $resultedit1=mysql_query("select
    memiliki_solusi.ID_solusi
    from virus_komputer,
    memiliki_solusi
    where
    memiliki_solusi.ID_virus=virus_komputer.ID_virus and
    virus_komputer.ID_virus='".$HTTP_GET_VARS["id"]."') or
    die(mysql_error());
  ?>

```


Pada modul *description list* akan menampilkan semua data – data virus yang ditambahkan dalam bentuk tabel, yaitu *virus name*, *target*, *active time* dan *solution*.

4.1.2.7 Modul Logout

Pada menu *logout* ini digunakan untuk administrator untuk keluar dari sistem ini. Pada saat administrator menekan tombol *logout* maka secara otomatis administrator keluar dari sistem dan akan ditampilkan pesan sebagai berikut :

“You had been logged out from the system”.

4.1.3 Modul PHP membuat halaman XML

Pada sistem ini PHP akan membuat halaman XML yang dapat panggil pada *internet explorer* dan aplikasi Visual Basic. Halaman XML pada *internet explorer* dapat dipanggil dengan memanggil *file xml_vy.php*.

Pada halaman untuk membuat XML ini, XML diolah di dalam mesin PHP sehingga menghasilkan kode XML. Pada dokumen XML yang dibuat merupakan jenis dokumen XML validasi, jadi pada dokumen XML yang dibuat melibatkan DTD sebagai pendeklarasian untuk validasi. DTD ini akan mengecek apakah elemen – elemen data yang dibuat sesuai dengan validasi DTD yang telah didefinisikan, jika sesuai berarti pengecekan berhasil tetapi jika tidak sesuai maka akan ditampilkan pesan *error* karena data tidak *valid*.

Pada pembuatan program XML dan validasinya (DTD), langkah awal yang perlu dilakukan adalah memanggil *file connet.php* yang telah dibuat sebelumnya yang akan digunakan untuk mengkoneksikan *database MySQL*.

dengan PHP. Sintaks yang digunakan untuk memanggil *file connet.php* adalah sebagai berikut :

```
include("connection/connect.php");
```

Setelah itu akan dilakukan penginisialisasi dokumen XML terlebih dahulu, sintaks programnya dapat dilihat sebagai berikut :

```
$doc = new_xmldoc("1.0");
```

Pada sintaks tersebut digunakan untuk menandai bahwa versi XML yang menggunakan adalah XML versi 1.0.

Dalam modul untuk membuat XML, XML akan diolah dalam mesin PHP maka dari itu akan ditampilkan halaman XML versi PHP. Agar tampilan yang dihasilkan adalah halaman XML dalam format XML maka akan digunakan sintaks sebagai berikut :

```
header("Content-Type: text/xml");
header("Pragma: no-cache");
```

Pembuatan *root node* ALLData dalam program ini menggunakan kode program sebagai berikut :

```
$root = $doc->add_root("ALLData");
```

Kode program tersebut berarti membuat *root node* dengan nama ALLData.

Pembentukan *child virus_komputer* dan *subchild* yang ada di dalamnya dibuat dengan cara melakukan *query database* untuk *record – record*. Kode program yang digunakan dalam *query database* adalah sebagai berikut :

```
$query_cv = "SELECT ID_virus, nama_virus, keterangan_virus
FROM virus_komputer order by ID_virus asc";
$result_cv = mysql_query($query_cv) or die ("Error in query:
$query_cv. " . mysql_error());
while($row1 = mysql_fetch_object($result_cv))
{
    $record_cv = $root->new_child("virus_komputer", "");
    $record_cv->set_attribute("ID_virus",$row1->ID_virus);
    $record_cv->new_child("nama_virus",$row1->nama_virus);
    $record_cv->new_child("keterangan_virus",$row1-
>keterangan_virus);
```



Query yang digunakan untuk menyeleksi semua data dalam tabel virus komputer adalah sebagai berikut :

```
$query_cv = "SELECT ID_virus, nama_virus, keterangan FROM
virus_komputer order by ID_virus asc";
```

Hasil dari *query* akan disimpan dalam variabel `$result_cv`. Kode program yang digunakan untuk membentuk *child* virus_komputer adalah sebagai berikut :

```
$record_cv = $root->new_child("virus_komputer", "");
```

Pada pembentukan *child* virus_komputer, ID_virus akan diset sebagai atribut.

Kode program yang digunakan untuk mengeset ID_virus sebagai atribut adalah sebagai berikut :

```
$record_cv->set_attribute("ID_virus", $row1->ID_virus);
```

Pada pembentukan *child* serang dan *subchild* yang ada di dalamnya, kode program yang digunakan adalah sebagai berikut :

```
$ID_virus_parent = $row1->ID_virus;

$query_srg = "SELECT bulan_aktif_virus, tempat_serang FROM
serang, sasaran_serang where (serang.ID_serang =
sasaran_serang.ID_serang) AND (serang.ID_virus =
$ID_virus_parent)";
$result_srg = mysql_query($query_srg) or die ("Error in query:
$query_srg. " . mysql_error());
$record_srg = $record_cv->new_child("serang", "");
while($row2 = mysql_fetch_object($result_srg))
{
    $record_def = $record_srg->new_child("detail_serang", "");
    $record_def->new_child("bulan_aktif_virus", $row2-
>bulan_aktif_virus);
    $record_def->new_child("tempat_serang", $row2-
>tempat_serang);
}
}
```

Pada pembentukan *child* solusi dan *subchild* yang ada di dalamnya, kode program yang digunakan adalah sebagai berikut :

```

$query_sol = "SELECT solusi FROM solusi,memiliki_solusi where
(solusi.ID_solusi = memiliki_solusi.ID_solusi) AND
(memiliki_solusi.ID_virus = $ID_virus_parent)";
$result_sol = mysql_query($query_sol) or die ("Error in query:
$query_sol. " . mysql_error());
$record_sol = $record_cv->new_child("solusi_virus", "");
while($row3 = mysql_fetch_object($result_sol))
{
    $record_sol->new_child("keterangan_solusi", $row3->solusi);
}

```

Pada modul untuk membuat halaman XML, sintaks yang digunakan untuk menampilkan semua data secara lengkap akan disimpan dalam suatu variabel yang kemudian akan ditampilkan. Sintaks yang digunakan adalah sebagai berikut :

```
$strXML = $doc->dumppmem();
```

Sintaks DTD yang dipakai dalam program akan disimpan dalam suatu variabel yang nantinya akan ditampilkan, kode program yang digunakan adalah sebagai berikut :

```

$dtd = "
<!DOCTYPE ALLData [
    <!ELEMENT ALLData (virus_komputer)+>
    <!ELEMENT virus_komputer (nama_virus, keterangan_virus,
    serang, solusi_virus)>
    <!ELEMENT nama_virus (#PCDATA)>
    <!ELEMENT keterangan_virus (#PCDATA)>
    <!ELEMENT serang (detail_serang)*>
    <!ELEMENT detail_serang (bulan_aktif_virus,
    tempat_serang)>
    <!ELEMENT bulan_aktif_virus (#PCDATA)>
    <!ELEMENT tempat_serang (#PCDATA)>
    <!ELEMENT solusi_virus (keterangan_solusi)*>
    <!ELEMENT keterangan_solusi (#PCDATA)>
    <!ATTLIST virus_komputer ID_virus CDATA '0'>
]>";

```

Pada kode program pembentukasn DTD tersebut penjelasannya adalah sebagai berikut :

1. Sintaks `<!DOCTYPE ALLData` menyatakan elemen puncak.

2. Sintaks `<!ELEMENT ALLData (virus_komputer)+>` menyatakan elemen `ALLData` memiliki elemen `virus_komputer` dan tanda `+` menyatakan bahwa `virus_komputer` bias berisi 1...N.

3. Sintaks `<!ELEMENT virus_komputer (nama_virus, keterangan_virus, serang, solusi_virus)>` menyatakan elemen `virus_komputer` memiliki *subchild* `nama_virus`, `keterangan_virus`, `serang` dan `solusi_virus`.

4. Pendeklarasian data dalam elemen `virus_komputer` menggunakan sintaks berikut :

```
<!ELEMENT nama_virus (#PCDATA)
<!ELEMENT keterangan_virus (#PCDATA)>
```

5. Sintaks `<!ELEMENT detail_serang(bulan_aktif_virus,tempat_serang)>` menyatakan elemen `detail_serang` memiliki elemen `bulan_aktif_virus` dan tempat `serang`.

6. Pendeklarasian data dalam elemen `detail_serang` menggunakan sintaks berikut :

```
<!ELEMENT bulan_aktif_virus (#PCDATA)>
<!ELEMENT tempat_serang (#PCDATA)>
```

7. Sintaks `<!ELEMENT solusi_virus (keterangan)*>` menyatakan elemen `solusi_virus` memiliki elemen `keterangan` dan tanda `*` menyatakan bahwa `keterangan` bisa berisi 0,1..N.

8. Pendeklarasian data dalam elemen `solusi-virus` adalah sebagai berikut :

```
<!ELEMENT keterangan (#PCDATA)
```

9. Sintaks `<!ATTLIST virus_komputer ID_virus CDATA '0'>` menyatakan bahwa `ATTLIST` digunakan untuk mengeset atribut di dalam `virus_komputer`. `CDATA` menyatakan bentuk `ID_virus` dalam bentuk karakter dan tanda `'0'`

menyatakan bahwa ID_virus yang ada akan ditampilkan seperti yang telah dimasukkan.

4.1.4 Modul Antar Muka Pihak Client

Pada modul antar muka pada pihak *client* ini akan disimpan dalam *projectvirus.vbp* yang di dalamnya terdapat beberapa *form*, untuk lebih jelasnya akan dibahas satu persatu tampilan *form* yang ada.

4.1.4.1 Modul untuk Tampilan Awal

Pada modul tampilan awal pada pihak *client* ini (*Load.frm*) digunakan untuk membuat *form* tampilan awal. Pada *form* tampilan awal ini akan ditampilkan logo dari Sanata Dharma dua tombol, yaitu tombol *next* dan *close*. Tombol *next* digunakan untuk *link* ke halaman berikutnya dan tombol *close* untuk keluar dari sistem apabila tidak jadi masuk ke dalam sistem.

Sintaks yang digunakan untuk menampilkan menu_utama jika diklik tombol *next* adalah sebagai berikut :

```
Private Sub Image2_Click()
    Unload Me
    Menu_Utama.Show
End Sub
```

Sintaks yang digunakan untuk membatalkan masuk ke sistem jika diklik tombol X adalah sebagai berikut :

```
Private Sub Image3_Click()
    Unload Me
End Sub
```

4.1.4.2 Modul Menu Utama

Pada modul menu utama (*Menu_Utama.frm*) ini berfungsi untuk menampilkan menu – menu. Adapun menu yang ada dalam menu utama

adalah menu *File* dan menu *Help*. Pada menu *File* memiliki sub menu, yaitu submenu *setup*, *information of computer virus* dan *exit*. Sub menu *setup* ini berfungsi untuk memanggil alamat URL tempat *file* data – data virus komputer diletakkan. Submenu *information of virus komputer* berfungsi untuk menampilkan data – data virus komputer dalam bentuk *treeview* dan *text*. Pada submenu *information of virus komputer* ini dilengkapi dengan fasilitas *search*, yang digunakan untuk mencari data – data virus komputer berdasarkan *virus_name*, *active_time*, dan *target*. Pada menu *Help* terdapat sub menu *about programmer* dan *about program*. Sub menu *about programmer* berfungsi untuk menampilkan identitas pembuat program dan sub menu *about program* berfungsi untuk menampilkan keterangan jalannya sistem pada *client*.

Inti program untuk modul menu utama ini adalah sebagai berikut :

```
'fungsi Menu Utama
Private Sub Menu_Utama_Load()
On Error Resume Next
IOCV.Enabled = False
End Sub
'ketika setup dipilih form setup diload dan ditampilkan
Private Sub Setup_Click()
Load frmSetup
frmSetup.Show
IOCV.Enabled = False
End Sub

'jk IOCv di select, form virus ditampilkan
Private Sub IOCv_Click()
frmVirus.Show
End Sub

Private Sub Exit_Click()
Unload Me
End Sub

'load form about program di set center
Private Sub AboutProgram_Click()
frmAboutProgram.Left = Menu_Utama.Width / 2 -
frmAboutProgram.Width / 2
frmAboutProgram.Top = Menu_Utama.Height / 2 -
frmAboutProgram.Height / 2
frmAboutProgram.Show
End Sub
```

```
'load form about programmer di set center
Private Sub AboutProgrammer_Click()
    Load frmAboutProgrammer
    frmAboutProgrammer.Left = Menu_Utama.Width / 2 -
frmAboutProgrammer.Width / 2
    frmAboutProgrammer.Top = Menu_Utama.Height / 2 -
frmAboutProgrammer.Height / 2
    frmAboutProgrammer.Show
```

4.1.4.3 Modul Setup

Pada modul *setup (Setup.frm)* ini berfungsi untuk memanggil *file* tempat data – data virus komputer diletakkan. Pada *form setup* akan terdapat *text box* untuk menuliskan letak tempat data – data virus komputer diletakkan. *Text box* ini tidak boleh kosong, alamat URL harus diisikan. Jika alamat URL tidak diisikan dan langsung mengklik ok maka akan ada pesan *error* agar *text box* tersebut diisikan alamat URL. Jika *text box* diisikan dan alamat URL benar maka akan ditampilkan *form Virus*. Pada pengisian alamat pada *text box* ini, berarti data virus diambil secara *online*. Kode program yang digunakan untuk pengecekan *text box* sekaligus pengambilan data virus secara *online* adalah sebagai berikut :

```
Private Sub CmdOk_Click()
    If TxtUrl.Text = "" Then
        'url/document xml path kosong
        MsgBox "XML URL/Document must not empty!", vbCritical,
        "Setup Error"
    Else
        'url set 1 (gagal)
        url = Trim(TxtUrl.Text)
        If frmVirus.ParseXMLDoc(TxtUrl.Text) = 1 Then
            MsgBox "Load URL/Document failed!", vbCritical, "Error
            Load file"
        Else
            'selain kejadian di atas berarti 0 (sukses)
            If URLType = "" Then
                URLType = "Online"
            End If
            frmVirus.Header1.Caption = "The Information of
            Computer Virus (" & URLType & ")"
            frmVirus.Load = True           'formVirus diload
            frmSetup.Visible = False      'formSetup dist false
            frmSearch.Visible = False     'formSearch diset false
            Menu_Utama.IOCV.Enabled = True 'Menu_Utama diset
            true/diaktifkan
        End If
    End If
```


Kode program yang digunakan untuk pengecekan alamat URL jika *text box* alamat URL dikosongkan adalah sebagai berikut :

```
If TxtUrl.Text = "" Then
  'url/document xml path kosong
  MsgBox "XML URL/Document must not empty!", vbCritical,
  "Setup Error"
```

Kode program yang digunakan untuk pengecekan alamat URL jika alamat yang dimasukkan salah adalah sebagai berikut :

```
'url set 1 (gagal)
url = Trim(TxtUrl.Text)
If frmVirus.ParseXMLDoc(TxtUrl.Text) = 1 Then
  'MsgBox "Load URL/Document failed!", vbCritical, "Error
  'Load file"
```

Jika *ParseXMLDoc* = 1 berarti gagal dan selain itu berarti sukses dan program akan menjalankan perintah sesudah *else*, yaitu :

```
'selain kejadian di atas berarti 0 (sukses)
If URLType = "" Then
  URLType = "Online"
End If
frmVirus.Header1.Caption = "The Information of Computer
Virus (" & URLType & ")"
frmVirus.Load = True      'formVirus diload
frmSetup.Visible = False 'formSetup dist false
frmSearch.Visible = False 'formSearch diset false
Menu_Utama.IOCV.Enabled = True 'Menu_Utama diset
true/diaktifkan
```

Kode program yang digunakan untuk menandai data virus secara *online* adalah sebagai berikut :

```
If URLType = "" Then
  URLType = "Online"
End If
frmVirus.Header1.Caption = "The Information of Computer
Virus (" & URLType & ")"
```

Pada *form setup* ini, terdapat *command dialogue* yang digunakan untuk mengambil data virus yang secara *local*, kode program yang digunakan untuk membentuk *command dialogue* adalah sebagai berikut :

```
'cmd untuk ambil file xml secara lokal
Private Sub cmdXMLFile_Click()
    Menu_Utama.cdlgMain.Flags = cdIOFNFileMustExist Or _
        cdIOFNHideReadOnly Or _
        cdIOFNPathMustExist
    'menfilter semua file yang berekstensi xml
    Menu_Utama.cdlgMain.Filter = "XML Files (*.xml)|*.xml"
    'buka jendela open
    Menu_Utama.cdlgMain.ShowOpen

    If Menu_Utama.cdlgMain.FileName <> "" Then
        TxtUrl.Text = Menu_Utama.cdlgMain.FileName
        URLType = "local"
    End If
End Sub
```

Sintaks yang digunakan untuk *filter file* XML adalah sebagai berikut :

```
Menu_Utama.cdlgMain.Filter = "XML Files (*.xml)|*.xml"
```

Sintaks yang digunakan untuk membuka jendela XML adalah sebagai berikut :

```
Menu_Utama.cdlgMain.ShowOpen
```

Kode program yang digunakan untuk pengecekan penulisan alamat URL secara *local* jika ternyata *text box* nya dikosongkan atau tidak diisi adalah sebagai berikut :

```
If Menu_Utama.cdlgMain.FileName <> "" Then
    TxtUrl.Text = Menu_Utama.cdlgMain.FileName
    URLType = "local"
End If
```

4.1.4.4 Modul *Information Of Virus Computer*

Modul *information of virus computer (Virus.frm)* digunakan untuk menampilkan data – data virus komputer yang telah *diload* dari alamat URL dalam bentuk *treeview*, *text* dan *grid*.

Deklarasi variabel yang digunakan pada modul virus sebagai berikut :

```
Public ParseValue           As Integer
Public ObjDOMDocument       As MSXML.DOMDocument
Public ObjXMLDOMNode        As MSComctlLib.Node
Public ObjXMLDOMNodeList    As IXMLDOMNodeList
Public Load                  As Boolean
Public NextKey               As Long
Public W, i, J, X, Y1, Y2, Y3, Sign    As Integer
Dim Result                  As Long
```

Keterangan dari masing – masing variabel adalah sebagai berikut :

1. *ParseValue* merupakan variabel yang digunakan pada seleksi *search* sesuai dengan *ListIndex* yang diset *integer*
2. *ObjDOMDocument* merupakan variabel yang digunakan untuk membentuk objek DOM yang diset *MSXML.DOMDocument* yang artinya objek yang digunakan untuk menrepresentasikan *root* dari *file XML*.
3. *ObjXMLDOMNode* merupakan variabel yang digunakan untuk merepresentasikan *node* di dalam dokumen *tree*.
4. *ObjXMLNodeList* merupakan variabel yang digunakan untuk mengakses semua koleksi *XMLDOMNode* yang diset *IXMLDOMNodeList* yang artinya objek DOM yang berisi *library NodeList*.
5. *Load* merupakan variabel yang digunakan untuk load *file* yang diset *boolean*.
6. *Nextkey* merupakan variabel yang digunakan untuk memgeset nilai atau *key* dari tiap *root* yang diset *long*.
7. *W, I, J, X, Y1, Y2, Y3, Sign* merupakan variabel yang digunakan dalam representasi *treview* ke dalam bentuk *grid*.
8. *Result* merupakan variabel yang digunakan dalam *search node* dalam XML mengenai hasil pengecekan *search* untuk ditampilkan sesuai yang dicari.

Dalam modul virus terdapat suatu fungsi yang digunakan untuk memparse dokumen XML, fungsi tersebut adalah sebagai berikut :

```
Public Function ParseXMLDoc(Path As String) As Long
    Dim FailValue           As Long
    Set ObjDOMDocument = New MSXML.DOMDocument

    tvwXMLText.Nodes.Clear

    ObjDOMDocument.async = False
    ObjDOMDocument.validateOnParse = True
    ObjDOMDocument.Load Path      'Load file yang actual
```

```

If ObjDOMDocument.parseError.errorCode <> 0 Then
    FailValue = 1

Else

    Set ObjXMLDOMNode = tvwXMLText.Nodes.Add(, , "ROOT", "XML
    Document Contents")
    Set ObjXMLDOMNodeList =
    ObjDOMDocument.selectNodes("//virus_komputer")
    WalkChildNodes ObjDOMDocument.childNodes, ObjXMLDOMNode
        J = 1
        W = 1
    WalkChildNodesToGrid ObjDOMDocument.childNodes
    End If
End Function

```

DOMDocument merupakan bagian yang pertama kali dipanggil maka dari itu perlu diset yang pertama kali, sintaks yang digunakan untuk set *DOMDocument* yang pertama kali adalah sebagai berikut :

```
Set ObjDOMDocument = New MSXML.DOMDocument
```

Setelah *DOMDocument* dipanggil maka akan dilakukan *clear text* atau membersihkan isi *text*, sintaks yang digunakan adalah sebagai berikut :

```
tvwXMLText.Nodes.Clear
```

Dalam fungsi *ParseXMLDoc*, *DOMDocument* yang dipanggil akan dipanggil secara *synchronous*, artinya semua data akan dibaca 100% atau keseluruhan dan setelah itu akan *diparsing*. Sintaks yang digunakan untuk set *synchronous* ini adalah sebagai berikut :

```
ObjDOMDocument.async = False
```

Setelah pengecekan dinyatakan sukses untuk *diload* maka akan diset *ObjXMLDOMNode* untuk menambah *node* dalam *treeview*, sintaks yang digunakan adalah sebagai berikut :

```
Set ObjXMLDOMNode = tvwXMLText.Nodes.Add(, , "ROOT", "XML
Document Contents")
```

Kode program yang digunakan untuk set *ObjXMLDOMDocument* adalah sebagai berikut :

```
Set ObjXMLDOMNodeList =
ObjDOMDocument.selectNodes("//virus_komputer")
```

Pada kode program set *ObjXMLDOMDocument* akan melakukan select terhadap *node* *virus_komputer*. Selanjutnya dalam fungsi *ParseXMLDoc*, akan memanggil fungsi *WalkChildNodes* yang digunakan untuk menampilkan semua *node* dalam sebuah *treeview*. Sintaks yang digunakan untuk menampilkan semua *node* dalam sebuah *treeview* adalah sebagai berikut :

```
WalkChildNodes ObjDOMDocument.childNodes, ObjXMLDOMNode
```

Selain ditampilkan dalam bentuk *treeview*, dokumen XML juga akan ditampilkan dalam bentuk *grid*, sintaks yang digunakan adalah sebagai berikut :

```
J = 1
W = 1
WalkChildNodesToGrid ObjDOMDocument.childNodes
```

Dalam fungsi *ParseXMLDoc* memanggil *textXMLText*, inti program *textXMLText* adalah sebagai berikut :

```
Private Sub tvwXMLText_Click()
Dim ObjSelNode As Node
'mengambil properti - properti di node yang dipilih
Set ObjSelNode = tvwXMLText.SelectedItem
txtTarget.Text = ""
txtname.Text = ""
txtExplain.Text = ""
txtSolution.Text = ""
'membersihkan isi textbox
cboactive.Clear

i = 0
PopulateVirusDetails ObjSelNode
End Sub
```

Inti program *tvwXMLText* digunakan untuk mengeset *treeview* XML dalam *text box*, sedangkan fungsi yang digunakan untuk merepresentasikan *treeview* XML ke

dalam *text box* adalah fungsi *WalkChildNodesTotext*. Sintaks yang digunakan untuk menampilkan data virus dari *treeview* ke dalam *text box* saat diklik *node virus_komputer* adalah sebagai berikut :

```
PopulateVirusDetails ObjSelNode
```

Sedangkan *populateVirusDetails* itu sendiri diambil dari inti program berikut :

```
Public Sub PopulateVirusDetails(ObjSelNode As Node)
    Dim ObjXMLDOMNodeList As IXMLDOMNodeList
    Dim ObjChildElement As MSXML.IXMLDOMNode
    Dim Text1 As String

    'jk ObjSelNodenya kosong maka secara otomatis diakhiri
    If ObjSelNode Is Nothing Then Exit Sub

    If Trim(ObjSelNode.Tag) <> "" Then
        If tvwXMLText.SelectedItem.Child Is Nothing Then
            Else
                Text1 = tvwXMLText.SelectedItem.Child.Text
            End If

        Set ObjXMLDOMNodeList =
        ObjDOMDocument.selectNodes("//ALLData//virus_komputer[na
        ma_virus= '" & Text1 & "'")
        ' panggil fungsi untuk ke text
        WalkChildNodesToText ObjXMLDOMNodeList
    End If
End Sub
```

Pada fungsi *ParseXMLDoc* juga memanggil fungsi *WalkChildNodes*, inti program *WalkChildNodes* itu sendiri adalah sebagai berikut :

```
Public Function WalkChildNodes(NodeList As
MSXML.IXMLDOMNodeList, _
    TreeViewParentNode As MSComctlLib.Node) As Long

    Dim ObjElement As MSXML.IXMLDOMNode
    Dim ObjAttribute As MSXML.IXMLDOMAttribute
    Dim NewNode As MSComctlLib.Node

    For Each ObjElement In NodeList
        If Not (ObjElement.nodeName = "#text") Then
            NextKey = NextKey + 1

            If NextKey = 2 Then
                'do nothing
            End If
        End If
    Next ObjElement
End Function
```

```

Else
Set NewNode =
tvwXMLText.Nodes.Add(TreeViewParentNode.Key, tvwChild, _
"key-" & CStr(NextKey))
If (ObjElement.childNodes.length > 1) Then
If ObjElement.nodeName = "virus_komputer" Then
If Not (ObjElement.Attributes Is Nothing) Then
For Each ObjAttribute In
ObjElement.Attributes
NewNode.Text = ObjElement.nodeName & "
(ID virus: " & ObjAttribute.Value & ")"
Next
End If
Else
NewNode.Text = ObjElement.nodeName
End If

Else
If ObjElement.nodeName = "serang" Or
ObjElement.nodeName = "solusi_virus" Then
NewNode.Text = ObjElement.nodeName
Else
NewNode.Text = ObjElement.Text
End If
End If
NewNode.Tag = ObjElement.Text

End If
If ObjElement.childNodes.length >= 1 Then
WalkChildNodes ObjElement.childNodes, NewNode
End If

End If
Next
End Function

```

4.1.4.5 Modul *Search*

Pada modul *search* (*Search.frm*), user dapat melakukan *searching* berdasarkan *virus_name*, *active_time* dan *target*. Sintaks yang digunakan untuk memilih *search* berdasarkan *virus_name* adalah sebagai berikut :

```

If (cbosearch.ListIndex = 0) Then
frmSearch.Label10.Caption = "Enter The Name Of Computer Virus
: "
frmSearch.Show
ParseValue = cbosearch.ListIndex

```

Sintaks yang digunakan untuk memilih *search* berdasarkan *active_time* adalah sebagai berikut :

```

ElseIf (cbosearch.ListIndex = 1) Then
frmSearch.Label10.Caption = "Enter The Month Of Aktive Time
: "
frmSearch.Show
ParseValue = cbosearch.ListIndex

```

Sintaks yang digunakan untuk memilih search berdasarkan *target* adalah sebagai berikut :

```

ElseIf (cbosearch.ListIndex = 2) Then
frmSearch.Label10.Caption = "Enter The Target Of Computer
Virus : "
frmSearch.Show
ParseValue = cbosearch.ListIndex

```

Pada *search* berdasarkan *virus_name* akan menampilkan nama – nama virus yang dicari sesuai dengan keinginan *user*, kode program yang digunakan untuk *search virus_name* ini adalah sebagai berikut :

```

Set ObjXMLDOMNode = tvwXMLText.Nodes.Add(, , "ROOT", "XML
Document Contents")

Set ObjXMLDOMNodeList =
ObjDOMDocument.selectNodes("//virus_komputer[nama_virus= '"
& Trim(frmSearch.txtsearch.Text) & "'")

WalkChildNodes ObjXMLDOMNodeList, ObjXMLDOMNode

```

Pada *search* berdasarkan *virus_name*, *ObjDOMDocument* akan melakukan *select* pada *node virus_komputer* berdasarkan *virus_name*, maka program akan melakukan perintah *search* sesuai dengan keinginan *user*.

Pada *search* berdasarkan *active_time* dan *target* menggunakan prinsip pencarian yang berbeda dengan prinsip pencarian berdasarkan *virus_name*. Pada *search* berdasarkan *active_time* dan *target* menggunakan prinsip *remove*, jadi program akan menjalankan perintah untuk mencari data virus yang dicari *user* dan jika ditemukan data yang dicari *user* maka program akan melakukan *remove* terhadap data yang tidak dicari. Fungsi yang digunakan untuk prinsip *search* ini adalah sebagai berikut :


```

Public Function WalkChildNodesForSearch(NodeList As
MSXML.IXMLDOMNodeList,
TreeViewParentNode As MSCComctlLib.Node) As Long

Dim ObjElement          As MSXML.IXMLDOMNode
Dim ObjAttributeMap     As IXMLDOMNamedNodeMap
Dim ObjAttribute        As MSXML.IXMLDOMAttribute
Dim NewNode             As MSCComctlLib.Node

For Each ObjElement In NodeList
    If Not (ObjElement.nodeName = "#text") Then
        NextKey = NextKey + 1
        Set NewNode =
tvwXMLText.Nodes.Add(TreeViewParentNode.Key, tvwChild, _
        "key-" & CStr(NextKey))

        If ObjElement.childNodes.length >= 1 Then
            WalkChildNodesForSearch ObjElement.childNodes, NewNode
        End If

        If (ObjElement.childNodes.length > 1) Then
            If ObjElement.nodeName = "virus_komputer" Then
                If Not (ObjElement.Attributes Is Nothing) Then
                    For Each ObjAttribute In ObjElement.Attributes
                        If Sign = 1 Then
                            NewNode.Text = ObjElement.nodeName & "
(ID virus: " & ObjAttribute.Value & ")"
                            Sign = 0
                        Else
                            tvwXMLText.Nodes.Remove NewNode.Index
                        End If
                    Next
                End If
            Else
                If Sign = 1 Then
                    NewNode.Text = ObjElement.nodeName
                End If
            End If
        Else
            If ObjElement.nodeName = "serang" Or
ObjElement.nodeName = "solusi_virus" Then
                NewNode.Text = ObjElement.nodeName
            Else
                Result = InStr(1, ObjElement.Text,
Trim(frmSearch.txtsearch.Text), vbBinaryCompare)
                If Result <> 0 Then
                    Sign = 1
                End If
                NewNode.Text = ObjElement.Text
            End If
        End If
        NewNode.Tag = ObjElement.Text
    End If
Next
End Function

```

4.1.4.6 Modul *Help About programmer*

Modul *help about programmer (Aboutprogrammer.frm)* hanya digunakan untuk menampilkan foto dan identitas pembuat program. Sintaks yang digunakan dalam modul *help about programmer* adalah sebagai berikut :

```
Private Sub Form_Load()  
    Image1.Picture = Menu_Utama.ImageList1.ListImages(6).Picture  
End Sub  
  
'jk di exit maka di hide  
Private Sub ExitButton2_Click()  
    Me.Hide  
End Sub
```

4.1.4.7 Modul *Help About Program*

Modul *help about program (Aboutprogram.frm)* hanya digunakan untuk menampilkan langkah – langkah jalannya program. Sintaks yang digunakan dalam modul *help about programmer* adalah sebagai berikut :

```
'jk exit maka di hide  
Private Sub ExitButton1_Click()  
    Me.Hide  
End Sub
```

BAB 5

ANALISIS IMPLEMENTASI

Secara garis besar yang akan dibahas dalam analisis implementasi ini adalah analisis manfaat, analisis teknologi, analisis karakteristik pengguna serta kelebihan dan kekurangan program.

5.1 Analisis Manfaat

Pada analisis ini akan dibahas tentang beberapa manfaat dibuatnya sistem aplikasi XML dalam pengaksesan *database server* dengan studi kasus *database virus komputer*. Pembahasan manfaat dalam pembuatan sistem aplikasi XML akan diperlihatkan pada Tabel 5.1.

Tabel 5.1 Manfaat pembuatan sistem aplikasi XML

No	Sudut Pandang	Manfaat
1	Pengaksesan data virus komputer dalam <i>database server</i> pada pihak <i>client</i> dengan aplikasi yang berbeda.	Dengan memanfaatkan XML, maka data virus komputer dalam <i>database server</i> dapat diubah ke dalam format XML yang dengan mudah diakses oleh pihak <i>client</i> sekalipun dengan aplikasi yang berbeda. Pada pihak <i>client</i> tinggal memanggil <i>file</i> dokumen XML untuk <i>diload</i> ke dalam aplikasi yang berbeda. Data virus komputer yang dipanggil tidak lagi dalam bentuk <i>database</i> tetapi

		dalam format XML.
2	Pertukaran data virus komputer.	Biasanya <i>database server</i> dengan aplikasi <i>client</i> seperti Visual Basic dikoneksikan secara langsung, misalnya dengan ODBC. ODBC biasanya digunakan untuk <i>local</i> , tetapi XML bisa digunakan untuk <i>multi network</i> . Dengan dimanfaatkan XML, diubahnya data dalam <i>database server</i> ke dalam bentuk dokumen XML yang kemudian <i>diload</i> ke aplikasi yang berbeda maka akan didapatkan pertukaran data secara mudah dan cepat.
3	Informasi.	Pada pihak <i>client</i> dapat mengakses data - data virus komputer sehingga didapatkan informasi data - data virus komputer yang <i>terupdate</i> .

5.2 Analisis Teknologi

Teknologi yang harus dipersiapkan untuk mengimplementasikan sistem yang dibuat dalam bentuk *client server* adalah sebagai berikut :

1. Meletakkan program yang terdapat pada bagian *server* secara *intranet*.

Semua program yang ada pada bagian *server* akan di *upload* secara *intranet*. Dalam sistem ini, bagian yang di *upload* secara *intranet* adalah semua modul untuk administrator, modul PHP membuat halaman XML dan *database server* MySQL. Hal – hal yang perlu diperhatikan dalam meletakkan aplikasi secara *intranet* adalah sebagai berikut :

- a. Membuat *user* untuk dapat meletakkan aplikasi secara *intranet*. Dalam sistem ini akan dibuat *user* *evy* dengan dilengkapi *password* , sehingga ada tempat untuk meletakkan aplikasi secara *intranet*.
- b. Setelah mendapatkan *user* maka dilanjutkan dengan meletakkan aplikasi ke *server*. Cara meletakkan aplikasi ke *server*, yaitu :
 - *User* akan melakukan ftp `://172.21.206.2`.
 - *User* akan melakukan *login*, setelah itu dilanjutkan dengan meletakkan aplikasi secara *intranet*.
- c. Membuat *user* untuk mendapatkan hak akses di pengaksesan *database server*, yaitu MySQL secara *intranet*. Dalam sistem ini akan dibuat *user* *evy* dengan dilengkapi *password*, sehingga bisa meletakkan *database virus* secara *intranet*. Alamat yang digunakan untuk meletakkan dan masuk dalam sistem *database server* MySQL, yaitu : http `://172.21.206.2/~MyAdmin` atau telnet `172.21.206.2`.
- d. Setelah aplikasi diletakkan ke *server* maka aplikasi bagian *server* ini bisa dipanggil dengan alamat, yaitu : http `://www.ti.usd/~evy/Virus` atau http `://172.21.206.2/~evy/Virus`.

2. Mengeset komputer *client* sehingga menjadi satu jaringan dengan komputer *server*. Hal – hal yang perlu diperhatikan dalam mengeset komputer *client*, yaitu :
 - a. Mengeset *ip address*, *subnet* dan *default gateway* pada *Network Neighborhood* pada bagian propertis sehingga menjadi satu jaringan dengan *server*.
 - b. Pada saat pengambilan data virus komputer secara *online* bagian *client* (Visual Basic), alamat URL nya adalah : `http://www.ti.usd/~evy/Virus/xml_evy.php` atau `http://172.21.206.2/~evy/Virus/xml_evy.php`.

5.3 Analisis Karakteristik Pengguna

Perangkat lunak untuk aplikasi XML dalam pengaksesan *database server* ini, dibuat untuk memudahkan *user* dalam melakukan pertukaran data antara pihak *server* dan *client* dengan aplikasi yang berbeda.

Sistem aplikasi XML ini terdapat dua pihak yang mempunyai peran penting, yaitu pihak *server* sebagai administrator dan pihak *client* sebagai *user*. Beberapa hal yang harus dipenuhi oleh pengguna dalam sistem ini untuk pihak *server* dan pihak *client* adalah sebagai berikut ini.

5.3.1 Analisis Karakteristik Pihak Server

Persyaratan yang harus dipenuhi pada pihak *server* sebagai administrator adalah sebagai berikut :

1. Minimal dapat mengoperasikan komputer dengan SO *Windows*.

2. Menguasai proses tambah, *edit* dan hapus data.
3. Menguasai PHP, *database* MySQL, XML dan Visual Basic untuk mengembangkan sistem.
4. Mengetahui perkembangan dunia virus komputer sehingga memudahkan dalam melakukan *update* data – data virus komputer.

5.3.2 Analisis Karakteristik Pihak *Client*

Persyaratan yang harus dipenuhi pada pihak *client* sebagai *user* adalah sebagai berikut :

1. Minimal dapat mengoperasikan komputer dengan SO *Windows*.
2. Memahami cara pengoperasian dengan aplikasi Visual Basic.
3. Memahami konsep pengloadan data dari *server* ke *client*.
4. Dapat melakukan proses *searching* data – data virus komputer.

5.4 Kelebihan dan kekurangan Program

5.4.1 Kelebihan Program

Kelebihan Program yang telah dibuat akan dibahas dengan membagi dua sisi yaitu pihak *server* sebagai administrator dan pihak *client* sebagai *user*.

1. Pihak *server* sebagai administrator.
 - a. Administrator dapat melakukan proses tambah data, edit data dan hapus data virus komputer.
 - b. Administrator dapat melihat isi data – data virus komputer melalui halaman *basic list* dan *description list*.

- c. Pada saat akan melakukan tambah data virus komputer, administrator dapat melihat terlebih dahulu mengenai data – data target serang dan solusi pada *target list* dan *solution list* sehingga jika target serang dan solusi virus sudah ada maka administrator hanya akan melakukan *add* virus pada halaman *add virus* dan mengenai target serang dan solusi virus dapat dilakukan pada halaman *description list*.
- d. Dengan menggunakan fasilitas *session* maka tidak sembarangan orang bisa mengakses halaman menu admin, karena sebelum masuk halaman administrator maka administrator harus memasukkan nama administrator dan *password*. Dan itu hanya diketahui oleh administrator saja.
- e. Program ini mengubah data dalam *database server* ke dalam format XML sehingga diperoleh suatu format umum yang dapat dipanggil pada *internet explore* ataupun pihak *client* dengan aplikasi yang berbeda, misalnya Visual Basic.

2. Pihak Client sebagai *user*

- a. *User* bisa mendapatkan informasi data – data virus komputer terbaru hanya dengan cara *ngload* data virus komputer yang telah dalam format xml, baik secara *online* maupun *local*.
- b. *User* mendapatkan informasi data – data virus komputer tidak lagi dalam bentuk *database* tetapi dalam bentuk format xml, jadi lebih mudah untuk diakses pada aplikasi yang berbeda.

- c. *User* dapat melakukan *searching* data – data virus komputer berdasarkan *virus_name*, *active_time* dan *target*.
- d. Aplikasi pada pihak *client* dilengkapi dengan menu *Help*, yang terdiri dari *about programmer* dan *about program*. *About programmer* akan memberikan informasi mengenai data – data pembuat program dan *about program* akan menampilkan mengenai petunjuk jalannya program.

5.4.2 Kekurangan Program

Selain mempunyai kelebihan program, sistem aplikasi XML ini memiliki kekurangan program pada sistem yang dibuat. Kekurangan yang terdapat dalam sistem aplikasi XML yang telah dibuat adalah sebagai berikut :

- a. Sistem aplikasi XML ini tidak memperhatikan *security* dalam pertukaran data, karena sistem ini hanya memperlihatkan cara pertukaran data dalam format dokumen XML sehingga mudah diakses pada aplikasi yang berbeda.
- b. Data – data virus komputer yang *diinputkan* sangat terbatas karena keterbatasan terhadap *resources* data – data virus komputer.

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Beberapa kesimpulan yang dapat diambil dari pembuatan sistem aplikasi XML dalam pengaksesan *database server* dengan studi kasus *database virus komputer* ini adalah sebagai berikut :

1. Program PHP dan *database MySQL* dalam sistem ini digunakan untuk menampilkan data virus komputer pada halaman administrator. Data – data yang ditambahkan, *diedit* ataupun dihapus akan ditampung dalam *database MySQL* pada *server*.
2. Program PHP digunakan untuk mengubah data virus komputer dalam *database server* ke dalam format dokumen XML sehingga dapat *download* pada aplikasi yang berbeda. Data – data virus komputer yang ditampilkan pada pihak *client* bukan lagi dalam bentuk *database* tetapi dalam format XML. Dokumen XML ini juga dapat dipanggil pada *browser* dengan tampilan format XML.
3. Dengan dibuat dan dicobanya sistem aplikasi XML dalam pengaksesan *database virus komputer* ini, dapat dilihat pertukaran data antara pihak *server* dan *client* dengan aplikasi yang berbeda dapat dilakukan dengan mudah dengan memanfaatkan aplikasi XML.
4. Sistem ini dapat diterapkan dengan pertukaran data secara *intranet* maupun *internet*. Jadi apabila ada data dalam *database server* memiliki

client di berbagai tempat, kota maupun negara yang menggunakan aplikasi yang berbeda maka pertukaran data dapat memanfaatkan sistem aplikasi XML ini. Sistem aplikasi XML ini akan memudahkan pihak *server* dan *user* pada *client* dengan aplikasi yang berbeda karena bentuk pertukaran data mudah dan cepat.

6.2 Saran - Saran

Beberapa saran yang digunakan untuk meningkatkan kehandalan sistem aplikasi XML ini adalah sebagai berikut :

1. Pada pembuatan sistem aplikasi XML ini tidak memperhatikan *security* dalam pertukaran data, karena sistem ini hanya memperlihatkan bentuk pertukaran data dari *database server* ke dalam format dokumen XML sehingga dapat diakses pada aplikasi yang berbeda. Maka dari itu perlu adanya penerapan *security* data sehingga data – data yang ada tidak disalahgunakan oleh orang – orang yang tidak bertanggung jawab.
2. *Back up* data sebaiknya selalu dilakukan oleh pihak administrator mengingat data – data virus komputer ini penting, karena akan digunakan untuk pihak *client* sebagai informasi data – data virus komputer.
3. *Resources* mengenai data – data virus komputer perlu digali lagi agar data – data yang akan diakses pada pihak *client* lebih banyak sehingga lebih informatif.
4. Sistem yang dibangun menggunakan DTD (*Document Type Definition*) untuk pendeklarasian tipe data yang ada dalam dokumen XML, untuk



lebih baik lagi maka sebaiknya sistem menggunakan *schema* dalam pendeklarasian tipe data dalam dokumen XML.

5. Pada bagian *server*, halaman administrator untuk *input* data virus komputer disarankan adanya iden.ifikasi *virus name*, *target* dan *solution* agar tidak ada data yang sama dalam *input* data.
6. Pada bagian *client*, fasilitas *searching* disarankan untuk ditambahkan *searching* berdasarkan tiga kriteria sekaligus, yaitu berdasarkan *virus_name*, *active time* dan *target* agar *searching* lebih efektif.

DAFTAR PUSTAKA

Hartono, Joyianto HM, Akt, MBA, Ph.D. *Analisis & disain sistem informasi : pendekatan teori dan praktek aplikasi bisnis*, Yogyakarta : Andi Offset Yogyakarta, 2001.

Kurniawan, Yahya, ST. *Aplikasi Web Database dengan PHP dan MySQL*, Jakarta : PT. Elex Media Komputindo, 2002.

Kadir, Abdul. *Pemrograman Web Dinamis Menggunakan PHP*, Yogyakarta : Andi Offset Yogyakarta, 2002.

Wikanta, Prasaja. *Buku Pintar Internet Mendesain Web dengan Pemrograman XML*, Jakarta : PT.Elex Media Komputindo, 2001.

Wilson, Mark and Wilson, Tracey. *XML Programing With VB AND ASP*, United States of America : United States of America, 2000.

[http : //ilmucomputer.com/Tutorial Windows dan Internet/internet.pdf](http://ilmucomputer.com/Tutorial Windows dan Internet/internet.pdf).

[http : //ilmucomputer.com/Dasar Pemrograman PHP dan MySQL/phpmysql.pdf](http://ilmucomputer.com/Dasar Pemrograman PHP dan MySQL/phpmysql.pdf).

[http : //ilmucomputer.com/Pengantar XML/xml.pdf](http://ilmucomputer.com/Pengantar XML/xml.pdf).

[http : //www.PerfectXml.com](http://www.PerfectXml.com).

[http : //www.w3school.com](http://www.w3school.com).

[http : //www.vbxml.com](http://www.vbxml.com).

[http : //www.w3.org/DOM/activity](http://www.w3.org/DOM/activity).

[http : //www.chilkatsoft.com](http://www.chilkatsoft.com).

[http : //sourceforge.net/projects/phptriad](http://sourceforge.net/projects/phptriad).

[http : //www.php.net](http://www.php.net).

[http://www.lecturer.ukdw.ac.id/Budi Susanto](http://www.lecturer.ukdw.ac.id/Budi%20Susanto).

[http://www.mahasiswa.com/informasi/aLin dan TaHu/Virus](http://www.mahasiswa.com/informasi/aLin%20dan%20TaHu/Virus).

[http://www.neotek.com/Mengenal Infeksi Digital : Virus Komputer/virus.pdf](http://www.neotek.com/Mengenal%20Infeksi%20Digital%20:%20Virus%20Komputer/virus.pdf).

LAMPIRAN

Halaman untuk Server

Modul connect.php

```
<?
$host=mysql_connect("localhost", "", "") or die(mysql_error());
mysql_select_db("virus", $host) or die(mysql_error());
?>
```

Modul checkfirst.php

```
<?
if(isset($_HTTP_POST_VARS["login"])){
    $result=mysql_query("select 1 from admin
    where nama_admin='".$_$_HTTP_POST_VARS["nama"]."' and
    password='".$_$_HTTP_POST_VARS["password"]."'") or
    die(mysql_error());
    $row=mysql_fetch_row($result);
    if($row[0]==1){
        session_register("nama");
        session_register("password");
        $_HTTP_SESSION_VARS["nama"]=$_HTTP_POST_VARS["nama"];
        $_HTTP_SESSION_VARS["password"]=$_HTTP_POST_VARS["passwo
        rd"];
        echo"<metahttpequiv=\"refresh\"content=\"0;
        URL=home.php\">";
        exit();
    }
}
```

Modul checknext.php

```
<?
if(isset($_HTTP_SESSION_VARS["nama"])&&isset($_HTTP_SESSION_VARS["pa
ssword"])){
    $result=mysql_query("select 1 from admin where
    nama_admin='".$_$_HTTP_SESSION_VARS["nama"]."' and
    password='".$_$_HTTP_SESSION_VARS["password"]."'");
    $row=mysql_fetch_row($result);
    if($row[0]!='1'){
        echo "<meta http-equiv=\"refresh\"
        content=\"0;URL=index.php\">";
        exit();
    }
else{
    echo "<meta http-equiv=\"refresh\"
    content=\"0;URL=index.php\">";
    exit();
}
?>
```

Modul fungsi.php

```
<?
function print_header(){
```



```

?>

<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
<link href="kelas.css" rel="stylesheet" type="text/css">
<STYLE fprolloverstyle>A: hover {
    COLOR: #4C7088;
    text-decoration: underline;
}
</STYLE>
</head>

<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<center>
<TABLE WIDTH=700 BORDER=0 CELLPADDING=0 CELLSPACING=0>
  <TR>
    <TD>
      <IMG SRC="images/banneratas_02.gif" WIDTH=700 HEIGHT=117
      ALT=""></TD>
    </TR>
  </TABLE>

  <table width="700" height="68%"
  background="images/garistengah_15.gif" BORDER=0 CELLPADDING=0
  CELLSPACING=0>
    <tr>
      <td align="center" valign="middle">

<?
}
function print_footer(){
?>
      </td>
    </tr>
  </table>

  <table width="700" height=66 BORDER=0 CELLPADDING=0 CELLSPACING=0>
    <tr>
      <td></td>
    </tr>
  </table>

</center>
</body>
</html>
<?
}
?>

<?
function checknext(){

```

```

if (isset($HTTP_SESSION_VARS["nama"]) && isset($HTTP_SESSION_VARS["password"])) {
    $result=mysql_query("select 1 from admin where
nama_admin='".$HTTP_SESSION_VARS["nama"]."' and
password='".$HTTP_SESSION_VARS["password"]."'");
    $row=mysql_fetch_row($result);
    if ($row[0]!=1) {
        echo "<meta http-equiv=\"refresh\"
content=\"index.php\">";
        exit();
    }
}
else{
    echo "<meta http-equiv=\"refresh\"
content=\"index.php\">";
    exit();
}
}
?>

```

Modul navigasi.php

```

<?php
if (!isset($HTTP_GET_VARS['hal']))
    $hal=1;
else $hal=$HTTP_GET_VARS['hal'];

function init_print_nav($ _max="2") {
    global $hal, $limit;
    define (" _MAX", "$ _max");
    $dr=($hal-1)* _MAX; //untuk query item file induk
    $limit=" LIMIT $dr, " . _MAX;
}

function print_nav($thispage, $table, $id_name, $where=""){
    global $hal;
    if ($where!=""){
        $where="WHERE $where";
    }
    $result = mysql_query("SELECT count($id_name) FROM $table
$where");
    $total=mysql_fetch_row($result);
    $total[0];
    $jml_hal=ceil($total[0] / _MAX);
    $enter=0;

    if (!eregi("\?", $thispage) )
        $thispage.="?";
    else $thispage.="&";

    if ($hal!=1) {
        $backnext=$hal-1;
        echo "&laquo; <A
href=".$thispage."hal=$backnext>Previous</a>\n";
    }

    for ($i=1;$i<=$jml_hal;$i++) {

```

```

    $enter++;
    if ($enter==30) {
        echo "<BR>\n";
        $enter=0;
    }
    if ($hal==$i) echo "&nbsp;$i";
    else {
        echo "&nbsp;<A
        HREF=".$thispage."hal=$i>$i</A>\n";
    } //end of else
} //end of for

if ($hal!=$jml_hal && $jml_hal!=0){
    $backnext=$hal+1;
    echo " <A HREF=".$thispage."hal=$backnext>Next</A>
    &raquo;\n";
}
}
?>

```

Modul index.php

```

<?
session_start();
include "connection/connect.php";
include "includes/checkfirst.php";
include "includes/fungsi.php";

print_header();
?>
<table align="center">
  <tr>
    <td background="images/formlogin_18.gif" width="250"
    height="149" align="center">
      <table width="90%" class="tekshitam">
        <tr>
          <td align="center">
            Enter your name and passwd:
          </td>
        </tr>
      </table>
      <table width="90%" class="tekshitam">
        <form action="" method="post">
        <tr>
          <td>
            Name
          </td>
          <td>
            <input type="text" name="nama" class="textbox">
          </td>
        </tr>
        <tr>
          <td>
            Password
          </td>
          <td>

```

```

<input type="password" name="password" class="textbox">
</td>
</tr>
<tr>
    <td>
    </td>
    <td>
        <input type="submit" name="login" value="Login"
            class="button">
    </td>
</tr>
</form>
</table>
</td>
</tr>
</table>

```

```

<?
print_footer();
?>

```

Modul home.php

```

<?
session_start();
include "connection/connect.php";
include "includes/fungsi.php";
include "includes/navigasi.php";

//update virus_komputer
$resultjml=mysql_query("select max(ID_virus) from
virus_komputer");
$rowjml=mysql_fetch_row($resultjml);

init_print_nav(5);

for($i=1;$i<=$rowjml[0];$i++){
    if(isset($_HTTP_POST_VARS["save1".$i])){
        mysql_query("update virus_komputer
            set nama_virus='".$_HTTP_POST_VARS["namavirus".$i]."',
            keterangan_virus='".$_HTTP_POST_VARS["keteranganvirus".
            $i]."'
            where
            ID_virus='".$_HTTP_POST_VARS["tmpidvirus".$i]."'") or
            die(mysql_error());
    }
}
if(isset($_HTTP_POST_VARS["delete1"])){
    if($_HTTP_POST_VARS["delete"]){
        while(list($key,$value)=each($_HTTP_POST_VARS["delete"])){
            mysql_query("delete from virus_komputer
                where ID_virus='".$value."'");
        }
    }
}
}

```

```

//update target
$resultjml=mysql_query("select max(ID_serang) from
sasaran_serang");
$rowjml=mysql_fetch_row($resultjml);

for($i=1;$i<=$rowjml[0];$i++){
    if(isset($HTTP_POST_VARS["save2".$i])){
        mysql_query("update sasaran_serang
        set tempat_serang='".$HTTP_POST_VARS["target".$i]."'
        where
        ID_serang='".$HTTP_POST_VARS["tmpidtarget".$i]."'") or
        die(mysql_error());
    }
}

if(isset($HTTP_POST_VARS["delete2"])){
    if($HTTP_POST_VARS["delete"]){
        while(list($key,$value)= each($HTTP_POST_VARS["delete"])){
            mysql_query("delete from sasaran_serang
            where ID_serang='".$value."'");
        }
    }
}

//update solution
$resultjml=mysql_query("select max(ID_solusi) from solusi");
$rowjml=mysql_fetch_row($resultjml);

for($i=1;$i<=$rowjml[0];$i++){
    if(isset($HTTP_POST_VARS["save3".$i])){
        mysql_query("update solusi
        set solusi='".$HTTP_POST_VARS["solusi".$i]."'
        where
        ID_solusi='".$HTTP_POST_VARS["tmpidsolusi".$i]."'") or
        die(mysql_error());
    }
}

if(isset($HTTP_POST_VARS["delete3"])){
    if($HTTP_POST_VARS["delete"]){
        while(list($key,$value) = each($HTTP_POST_VARS["delete"])){
            mysql_query("delete from solusi
            where ID_solusi='".$value."'");
        }
    }
}

$resultvirus=mysql_query("select * from virus_komputer $limit");
$resulttarget=mysql_query("select * from sasaran_serang $limit");
$resultsolusi=mysql_query("select * from solusi $limit");

print_header();
?>
<table width="80%">
    <tr>
        <td>
            <a href="advirus.php"
onClick="JavaScript:window.open('advirus.php', 'PageName',
'toolbar=no,location=no,directories=no,status=no,menubar=no,scroll

```

```

bars=no,resizable=no,width=650,height=350');return false;"
class="link">Add Virus</a>
    |
    <a href="addtarget.php"
onClick="JavaScript:window.open('addtarget.php', 'PageName',
'toolbar=no,location=no,directories=no,status=no,menubar=no,scroll
bars=no,resizable=no,width=600,height=250');return false;"
class="link">Add Target</a>
    |
    <a href="addsolution.php"
onClick="JavaScript:window.open('addsolution.php', 'PageName',
'toolbar=no,location=no,directories=no,status=no,menubar=no,scroll
bars=no,resizable=no,width=600,height=250');return false;"
class="link">Add Solution</a>
        &nbsp;
        &nbsp;
        &nbsp;
        &nbsp;
        <a href="home.php" class="link">Basic List</a>
    |
    <a href="description.php"
class="link">Decription List</a>
    </td>
    <td align="right">
        <a href="logout.php" class="link">Logout</a>
    </td>
</tr>
</table>
<br>
<table width="80%">
    <tr>
        <td class="judulhitam" align="center">
            Basic List
        </td>
    </tr>
</table>
<div align="center">
<?php
    print_nav("home.php", "virus_komputer", "id_virus", "");
?>
</div>

<form action="" method="post" enctype="multipart/form-data">
<table width="80%">
    <tr>
        <td class="judulhitam">
             Virus List
        </td>
    </tr>
</table>
<table bgcolor="#4C7088" width="80%" cellpadding="2"
cellspacing="1">
    <tr align="center" class="judul">

```

```

        <td>
            ID Virus
        </td>
        <td>
            Virus Name
        </td>
        <td>
            Explain
        </td>
        <td>
            Edit
        </td>
        <td>
            <input type="submit" name="delete1" value="Delete"
            class="button">
        </td>
    </tr>
    <?
    while($rowvirus=mysql_fetch_row($resultvirus)){
    ?>
    <tr valign="top" bgcolor="#C6DAE8" class="tekshitam">
        <td>
            <input type="text"
            name="idvirus<?=$rowvirus[0]?>"
            value="<?=$rowvirus[0]?>" size="7" disabled
            class="textbox">
            <input type="hidden"
            name="tmpidvirus<?=$rowvirus[0]?>"
            value="<?=$rowvirus[0]?>" size="7">
        </td>
        <td>
            <input type="text"
            name="namavirus<?=$rowvirus[0]?>"
            value="<?=$rowvirus[1]?>" <?

            if(!isset($HTTP_POST_VARS["edit1".$rowvirus[0]])){
                echo " disabled";
            }
            ?> class="textbox">
        </td>
        <td>
            <input type="text"
            name="keteranganvirus<?=$rowvirus[0]?>"
            value="<?=$rowvirus[2]?>" <?

            if(!isset($HTTP_POST_VARS["edit1".$rowvirus[0]])){
                echo " disabled";
            }
            ?> class="textbox">
        </td>
        <td align="center">
            <?

            if(isset($HTTP_POST_VARS["edit1".$rowvirus[0]])){
                ?>

```

```

        <input type="submit"
        name="save1<?=$rowvirus[0]?>" value="Save"
        class="button">
    <?
    }
    else{
    ?>
        <input type="submit"
        name="edit1<?=$rowvirus[0]?>" value="Edit"
        class="button">
    <?
    }
    ?>
</td>
<td align="center">
    <input type="checkbox" name="delete[]"
    value="<?=$rowvirus[0]?>">
</td>
</tr>
<?
}
?>
</table>

<br>
<table width="80%">
    <tr>
        <td class="judulhitam">
             Target List
        </td>
    </tr>
</table>
<table bgcolor="#4C7088" width="80%" cellpadding="2"
cellspacing="1">
    <tr align="center" class="judul">
        <td>
            ID Target
        </td>
        <td>
            Target
        </td>
        <td>
            Edit
        </td>
        <td>
            <input type="submit" name="delete2"
            value="Delete" class="button">
        </td>
    </tr>
    <?
    while($rowtarget=mysql_fetch_row($resulttarget)){
    ?>
    <tr valign="top" bgcolor="#C6DAE8" class="tekshitam">
        <td>

```



```

        <input type="text"
        name="idtarget<?=$rowtarget[0]?>"
        value="<?=$rowtarget[0]?>" size="7" disabled
        class="textbox">
        <input type="hidden"
        name="tmpidtarget<?=$rowtarget[0]?>"
        value="<?=$rowtarget[0]?>" size="7">
    </td>
    <td align="left">
        <pre><textarea name="target<?=$rowtarget[0]?>"
cols="50" rows="3" <?
        if(!isset($HTTP_POST_VARS["edit2".$rowtarget[0]])){
            echo " disabled";
        }
        ?>
class="textbox"><?=$rowtarget[1]?></textarea></pre>
    </td>
    <td>
    <?
        if(isset($HTTP_POST_VARS["edit2".$rowtarget[0]])){
            ?>
                <input type="submit"
                name="save2<?=$rowtarget[0]?>"
                value="Save" class="button">
            <?
                }
            else{
            ?>
                <input type="submit"
                name="edit2<?=$rowtarget[0]?>"
                value="Edit" class="button">
            <?
                }
            ?>
        </td>
        <td align="center">
            <input type="checkbox" name="delete[]"
            value="<?=$rowtarget[0]?>">
        </td>
    </tr>
    <?
    }
    ?>
</table>
<br>
<table width="80%">
    <tr>
        <td class="judulhitam">
             Solution List
        </td>
    </tr>
</table>

```

```

<table bgcolor="#4C7088" width="80%" cellpadding="2"
cellspacing="1">
  <tr align="center" class="judul">
    <td>
      ID Solution
    </td>
    <td>
      Solution
    </td>
    <td>
      Edit
    </td>
    <td>
      <input type="submit"
name="delete3" value="Delete" class="button">
    </td>
  </tr>
  <?
while($rowsolusi=mysql_fetch_row($resultsolusi)){
  <?
  <tr valign="top" bgcolor="#C6DAE8" class="tekshitam">
    <td>
      <input type="text"
name="idsolusi<?=$rowsolusi[0]?>"
value="<?=$rowsolusi[0]?>" size="7" disabled
class="textbox">
      <input type="hidden"
name="tmpidsolusi<?=$rowsolusi[0]?>"
value="<?=$rowsolusi[0]?>" size="7">
    </td>
    <td>
      <pre><textarea name="solusi<?=$rowsolusi[0]?>"
cols="50" rows="6" <?
      if(!isset($HTTP_POST_VARS["edit3".$rowsolusi[0]])){
        echo " disabled";
      }
      <?
class="textbox"><?=$rowsolusi[1]?></textarea></pre>
    </td>
    <td>
      <?
      if(isset($HTTP_POST_VARS["edit3".$rowsolusi[0]])){
        <?
          <input type="submit"
name="save3<?=$rowsolusi[0]?>" value="Save" class="button">
        <?
      }
      else{
        <?
          <input type="submit"
name="edit3<?=$rowsolusi[0]?>"
value="Edit" class="button">
        <?
      }
    </td>
  </tr>
}

```

```

        ?>
    </td>
    <td align="center">
        <input type="checkbox" name="delete[]"
            value="<?=$rowsolusi[0]?>">
    </td>
</tr>
<?
}
?>
</table>
</form>

<div align="center">
    <?php
        print_nav("home.php", "virus_komputer", "id_virus", "");
    ?>
</div>

<?
print_footer();
?>

```

Modul Description.php

```

<?
session_start();
include "connection/connect.php";
include "includes/fungsi.php";
include "includes/navigasi.php";

$table="virus_komputer";
//update virus_komputer
init_print_nav(5);

if(isset($_HTTP_POST_VARS["add"])){
    if($_HTTP_POST_VARS["target"]){
        while(list($key,$value) =
            each($_HTTP_POST_VARS["target"])){
            mysql_query("insert into serang
                values('".$_HTTP_POST_VARS["virus"]."', '".$_$value.
                "','".$_HTTP_POST_VARS['masaaktif'].$value.'')")
                or die(mysql_error());
        }
    }

    if($_HTTP_POST_VARS["solusi"]){
        while(list($key,$value) =
            each($_HTTP_POST_VARS["solusi"])){
            mysql_query("insert into memiliki_solusi
                values('".$_HTTP_POST_VARS["virus"]."', '".$_$value.
                ."'") or die(mysql_error());
        }
    }
}

```

```

if(isset($_HTTP_POST_VARS["edit"])){
    mysql_query("delete from serang where
    ID_virus='".$_$_HTTP_POST_VARS["virus"]."'");
    if($_HTTP_POST_VARS["target"]){
        while(list($key,$value) =
        each($_HTTP_POST_VARS["target"])){
            mysql_query("insert into serang
            values('".$_$_HTTP_POST_VARS["virus"]."', '".$_$_value.
            "', '".$_$_HTTP_POST_VARS['masaaktif'].$value].'")
            or die(mysql_error());
        }
    }

    mysql_query("delete from memiliki_solusi where
    ID_virus='".$_$_HTTP_POST_VARS["virus"]."'");
    if($_HTTP_POST_VARS["solusi"]){
        while(list($key,$value1) =
        each($_HTTP_POST_VARS["solusi"])){
            mysql_query("insert into memiliki_solusi
            values('".$_$_HTTP_POST_VARS["virus"]."', '".$_$_value1
            ."'") or die(mysql_error());
        }
    }
}

$resultvirus=mysql_query("select * from virus_komputer");
$resulttarget=mysql_query("select ID_serang,
left(tempat_serang,20),tempat_serang from sasaran_serang") or
die(mysql_error());
$resultsolusi=mysql_query("select ID_solusi,
left(solusi,30),solusi from solusi");

print_header();
?>
<table width="80%">
    <tr>
        <td>
            <a href="advirus.php"
onClick="JavaScript:window.open('advirus.php', 'PageName',
'toolbar=no,location=no,directories=no,status=no,menubar=no,scroll
bars=no,resizable=no,width=650,height=350');return false;"
class="link">Add Virus</a>
            |
            <a href="addtarget.php"
onClick="JavaScript:window.open('addtarget.php', 'PageName',
'toolbar=no,location=no,directories=no,status=no,menubar=no,scroll
bars=no,resizable=no,width=600,height=250');return false;"
class="link">Add Target</a>
            |
            <a href="addsolution.php"
onClick="JavaScript:window.open('addsolution.php', 'PageName',
'toolbar=no,location=no,directories=no,status=no,menubar=no,scroll
bars=no,resizable=no,width=600,height=250');return false;"
class="link">Add Solution</a>
            &nbsp;
            &nbsp;

```

```

        &nbsp;
        &nbsp;
        <a href="home.php" class="link">Basic List</a>
        |
        <a href="description.php" class="link">Decription
        List</a>
    </td>
    <td align="right">
        <a href="logout.php" class="link">Logout</a>
    </td>
</tr>
</table>
<br>
<table>
    <tr>
        <td class="judulhitam">
            Description List
        </td>
    </tr>
</table>
<?
if (isset($_HTTP_GET_VARS["id"])){
    $id=$_HTTP_GET_VARS["id"];
}
else{
    $id=0;
}
?>
<iframe src="target_solusi.php?id=<?=$id?>" width="95%"
height="400" frameborder="1" scrolling="auto" ></iframe>

<table width="95%">
    <tr>
        <td class="judulhitam">
             Virus target and its solution
        </td>
    </tr>
</table>
<table bgcolor="#4C7088" width="95%" cellpadding="2"
cellspacing="1">
    <tr align="center" class="judul">
        <td>
            Virus Name
        </td>

        <td> Target/Active Time</td>
        <td>
            Solution
        </td>
        <td>
            Edit
        </td>
    </tr>
<?

```

```

$result=mysql_query("select * from virus_komputer $limit")
or die(mysql_error());
while($row=mysql_fetch_row($result)){
?>
<tr valign="top" bgcolor="#C6DAE8" class="tekshitam">
  <td>
    <p>[
      <?=$row[0]?>
    ]<br>
      <?=$row[1]?>
    </p>
  </td>
  <td>
    <?
      $result1=mysql_query("select sasaran_serang.ID_serang,
sasaran_serang.tempat_serang,
serang.bulan_aktif_virus from virus_komputer,
sasaran_serang,
serang
where virus_komputer.ID_virus=serang.ID_virus and
sasaran_serang.ID_serang=serang.ID_serang and
virus_komputer.ID_virus='".$row[0]."' ) or
die(mysql_error());
while($row1=mysql_fetch_row($result1)){
  echo "[$row1[0]] <br>$row1[1]. <b>Active
Time</b>: $row1[2]<br>";
}
?>
</td>
<td>
  <?
    $result2=mysql_query("select solusi.ID_solusi,
solusi.Solusi
from virus_komputer,
memiliki_solusi,
solusi
where virus_komputer.ID_virus=memiliki_solusi.ID_virus
and
solusi.ID_solusi=memiliki_solusi.ID_solusi and
virus_komputer.ID_virus='".$row[0]."'") or
die(mysql_error());
while($row2=mysql_fetch_row($result2)){
  echo "[$row2[0]] <br>$row2[1]<br>";
}
?>
</td>
<td>
  <p><a href="description.php?id=<?=$row[0]?>"
class="link">Edit</a></p>
</td>
</tr>
<?
}
?>
</table>

```

```

<div align="center">
<?php
    print_nav("description.php", "solusi", "id_solusi", "");
    ?>
</div>
<br>
<?
print_footer();
?>

```

Modul target_solusi.php

```

<?
session_start();
include "connection/connect.php";
include "includes/checknext.php";
include "includes/navigasi.php";

$stable="virus_komputer";
//update virus_komputer
init_print_nav(5);

if(isset($_HTTP_POST_VARS["add"])){
    if($_HTTP_POST_VARS["target"]){
        while(list($key,$value) =
            each($_HTTP_POST_VARS["target"])){
            mysql_query("insert into serang
                values('".$_HTTP_POST_VARS["virus"]."', '".$_$value.
                "', '".$_HTTP_POST_VARS['masaaktif'].$value.'')")
            or die(mysql_error());
        }
    }

    if($_HTTP_POST_VARS["solusi"]){
        while(list($key,$value1) =
            each($_HTTP_POST_VARS["solusi"])){
            mysql_query("insert into memiliki_solusi
                values('".$_HTTP_POST_VARS["virus"]."', '".$_$value1.
                ."'") or die(mysql_error());
        }
    }
}

if(isset($_HTTP_POST_VARS["edit"])){
    mysql_query("delete from serang where
        ID_virus='".$_HTTP_POST_VARS["virus"].'");
    if($_HTTP_POST_VARS["target"]){
        while(list($key,$value) =
            each($_HTTP_POST_VARS["target"])){
            mysql_query("insert into serang
                values('".$_HTTP_POST_VARS["virus"]."', '".$_$value.
                "', '".$_HTTP_POST_VARS['masaaktif'].$value.'')")
            or die(mysql_error());
        }
    }
}

```

```

mysql_query("delete from memiliki_solusi where
ID_virus='".$$_HTTP_POST_VARS["virus"]."");
if($_HTTP_POST_VARS["solusi"]){
    while(list($key,$value1) =
    each($_HTTP_POST_VARS["solusi"])){
        mysql_query("insert into memiliki_solusi
        values('".$_HTTP_POST_VARS["virus"]."', '".$_.$value1
        ."'") or die(mysql_error());
    }
}

$resultvirus=mysql_query("select * from virus_komputer");
$resulttarget=mysql_query("select ID_serang,
left(tempat_serang,20),tempat_serang from sasaran_serang") or
die(mysql_error());
$resultsolusi=mysql_query("select ID_solusi,
left(solusi,30),solusi from solusi");

?>
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
<link href="kelas.css" rel="stylesheet" type="text/css">
</head>

<body bgcolor="#C6DAE8">
<?
if(isset($_HTTP_GET_VARS["id"])&&$_HTTP_GET_VARS["id"]!="0"){
?>
<table width="95%" class="tekshitam">
<form action="description.php" method="post"
enctype="multipart/form-data" target="_parent">
    <tr valign="top">
        <td>
            Virus
        </td>
        <td>
            <select name="virus" class="textbox">
                <?
                while($rowvirus=mysql_fetch_row($resultvirus)){
                    <?
                        <option value="<?=$rowvirus[0]?>" <?
                            if($_HTTP_GET_VARS["id"]==<?=$rowvirus[0]?>){
                                echo " selected ";
                            }
                        <?><?=$rowvirus[1]?></option>
                    <?
                }
            <?>
            </select>

```



```

</td>
<td>
    Target
</td>
<td>
<?
$i=1;
while($rowtarget=mysql_fetch_row($resulttarget)){
$resultedit=mysql_query("select serang.ID_serang
from virus_komputer,
serang
where
serang.ID_virus=virus_komputer.ID_virus and
virus_komputer.ID_virus='".$$_HTTP_GET_VARS["id"]."'")
or die(mysql_error());
?>
    <input type="checkbox"
value="<?=$rowtarget[0]?>" name="target[]" <?

    while($rowedit=mysql_fetch_row($resultedit)){
        if($rowedit[0]==$rowtarget[0])
            echo " checked ";
    }
    ?><?=$rowtarget[1]?>....
<?
$result=mysql_query("select * from serang
where
ID_virus='".$$_HTTP_GET_VARS["id"]."' and
ID_serang='".$rowtarget[0]."'") or
die(mysql_error());
$row=mysql_fetch_row($result);
?>
<br>
Active time: <input
name="masaaktif<?=$rowtarget[0]?>" value="<?=$row[2]?>"
class="textbox"><br>
<?
    $i++;
}
?>
</td>
<td>
    Solusi
</td>
<td>
<?
while($rowsolusi=mysql_fetch_row($resultsolusi)){
$resultedit1=mysql_query("select
memiliki_solusi.ID_solusi
from virus_komputer,
memiliki_solusi
where
miliki_solusi.ID_virus=virus_komputer.ID_virus
and
virus_komputer.ID_virus='".$$_HTTP_GET_VARS["id"
]."'") or die(mysql_error());

```

```

?>
  <input type="checkbox"
  value="<?=$rowsolusi[0]?>"
  alt="<?=$rowsolusi[2]?>" name="solusi[]" <?
  while($rowedit1=mysql_fetch_row($resultedit1)){
    if($rowedit1[0]==$rowsolusi[0])
      echo " checked ";
    }
  ?><?=$rowsolusi[1]?>....<br>
  <?
  }
?>
</td>
<td>
  <input type="submit" name="edit" value=" Edit "
  class="button">
</td>
</tr>
</form>
</table>
<?
}
else{
?>
<table width="95%" class="tekshitam">
<form action="description.php" method="post"
enctype="multipart/form-data" target="_parent">
  <tr valign="top">
    <td>
      Virus
    </td>
    <td>
      <select name="virus" class="textbox">
      <?
      while($rowvirus=mysql_fetch_row($resultvirus)){
        ?>
          <option
            value="<?=$rowvirus[0]?>"><?=$rowvirus[1]?
          ></option>
        <?
        }
        ?>
      </select>
    </td>
    <td>
      Target
    </td>
    <td>
      <!--
      <select name=f"virus">
        <option
          value="<?=$rowtarget[0]?>"><?=$rowtarget[1]?>...
        </option>
        <?

```

```

        <?
        while($rowtarget=mysql_fetch_row($resulttarget))
        {
        ?>
            <input type="checkbox"
            value="<?=$rowtarget[0]?>"
            name="target[]">
            <?=$rowtarget[1]?>....
            <br>
            Active time:
            <input name="masaaktif<?=$rowtarget[0]?>"
            class="textbox"><br>
        <?
        }
        ?>
    </td>
    <td>
        Solusi
    </td>
    <td>
        <?

        while($rowsolusi=mysql_fetch_row($resultsolusi)){
            ?>
                <input type="checkbox"
                value="<?=$rowsolusi[0]?>"
                alt="<?=$rowsolusi[2]?>"
                name="solusi[]"><?=$rowsolusi[1]?>....<br>
            <?
            }
            ?>
        </td>
        <td>
            <input type="submit" name="add" value=" Add "
            class="button">
        </td>
    </tr>
</form>
</table>
<?
}
?>
</body>
</html>

```

Modul addvirus.php

```

<?
session_start();
include "connection/connect.php";
include "includes/checknext.php";

$error="";

if(isset($_HTTP_POST_VARS["add"])){

```

```

if(!empty($_HTTP_POST_VARS["namavirus"])){
    mysql_query("insert into virus_komputer
        values('',
            '$_HTTP_POST_VARS["namavirus"].'',
            '$_HTTP_POST_VARS["keteranganvirus"].''')")
    };
}
else{
    $error="Error..... Virus Name must be fill...";
}
}
?>
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="refresh" content="index.php">
<link href="kelas.css" rel="stylesheet" type="text/css">
<script language="JavaScript1.2">
function closewindow(){
    window.close();
}
</script>
</head>

<body background="images/garistengah_15.gif">
<table width="600">
    <tr>
        <td class="judulhitam">
            Add Virus
        </td>
        <td align="right">
            <input type="button" name="close" value="Close
            window" onClick="closewindow()" class="button">
        </td>
    </tr>
    <tr>
        <td>
            <font color="#FF0000"><?=$error?></font>
        </td>
    </tr>
</table>
<table bgcolor="#4C7088" width="600" cellpadding="2"
cellspacing="1">
    <tr>
        <td class="judul">
            Add
        </td>
    </tr>
    <tr>
        <td bgcolor="#C6DAE8">
            <table width="600" class="tekshitam">
                <form action="" method="post">
                    <tr>
                        <td width="40%">
                            Virus Name

```

```

        </td>
        <td>
            <input type="text"
                name="namavirus"
                class="textbox">
        </td>
    </tr>
    <tr>
        <td valign="top">
            Explain
        </td>
        <td>
            <textarea
                name="keteranganvirus"
                cols="50" rows="8"
                class="textbox"></textarea>
        </td>
    </tr>
    <tr>
        <td>
        </td>
        <td>
            <input type="submit"
                name="add" value=" Add "
                class="button">
                &nbsp;
            <input type="reset"
                name="reset" value="Clear"
                class="button">
        </td>
    </tr>
</form>
</table>
</td>
</tr>
</table>
</body>
</html>

```

Modul addtarget.php

```

<?
session_start();
include "connection/connect.php";
include "includes/checknext.php";

$error="";

if(isset($_HTTP_POST_VARS["add"])){
    if(!empty($_HTTP_POST_VARS["target"])){
        mysql_query("insert into sasaran_serang
            values('',
                '$_HTTP_POST_VARS["target"].''");
    }
}

```

```

        else{
            $error="Error..... Target must be fill...";
        }
    }
    ?>
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="refresh" content="index.php">
<link href="kelas.css" rel="stylesheet" type="text/css">
<script language="JavaScript1.2">
function closewindow(){
    window.close();
}
</script>
</head>

<body background="images/garistengah_15.gif">
<table width="500">
    <tr>
        <td class="judulhitam">
            Add Target
        </td>
        <td align="right">
            <input type="button" name="close" value="Close
            window" onClick="closewindow()" class="button">
        </td>
    </tr>
    <tr>
        <td>
            <font color="#FF0000"><?=$error?></font>
        </td>
    </tr>
</table>
<table bgcolor="#4C7088" width="500" cellpadding="2"
cellspacing="1">
    <tr>
        <td class="judulhitam">
            Add
        </td>
    </tr>
    <tr>
        <td bgcolor="#C6DAE8">
            <table width="500" class="tekshitam">
                <form action="" method="post">
                    <tr>
                        <td width="30%">
                            Target
                        </td>
                        <td>
                            <textarea name="target"
                            cols="50" rows="3"
                            class="textbox"></textarea>
                        </td>
                    </tr>
                </form>
            </table>
        </td>
    </tr>
</table>

```

```

        <tr>
            <td>
                </td>
            </td>
            <td>
                <input type="submit"
                    name="add" value=" Add "
                    class="button">
                &nbsp;
                <input type="reset"
                    name="reset" value="Clear"
                    class="button">
            </td>
        </tr>
    </form>
</table>
</td>
</tr>
</table>
</body>
</html>

```

Modul addsolution.php

```

<?
session_start();
include "connection/connect.php";
include "includes/checknext.php";

$error="";

if(isset($_HTTP_POST_VARS["add"])){
    if(!empty($_HTTP_POST_VARS["target"])){
        mysql_query("insert into sasaran_serang
                    values('',
                        '".$_HTTP_POST_VARS["target"]."'");
    }
    else{
        $error="Error..... Target must be fill...";
    }
}
?>
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="refresh" content="index.php">
<link href="kelas.css" rel="stylesheet" type="text/css">
<script language="JavaScript1.2">
function closewindow(){
    window.close();
}
</script>
</head>

<body background="images/garistengah_15.gif">

```

```

<table width="500">
  <tr>
    <td class="judulhitam">
      Add Target
    </td>
    <td align="right">
      <input type="button" name="close" value="Close
      window" onClick="closewindow()" class="button">
    </td>
  </tr>
</table>
<table bgcolor="#4C7088" width="500" cellpadding="2"
cellspacing="1">
  <tr>
    <td class="judulhitam">
      Add
    </td>
  </tr>
  <tr>
    <td bgcolor="#C6DAE8">
      <table width="500" class="tekshitam">
        <form action="" method="post">

          <tr>
            <td width="30%">
              Target
            </td>
            <td>
              <textarea name="target"
              cols="50" rows="3"
              class="textbox"></textarea>
            </td>
          </tr>
          <tr>
            <td>
            </td>
            <td>
              <input type="submit"
              name="add" value=" Add "
              class="button">
              &nbsp;
              <input type="reset"
              name="reset" value="Clear"
              class="button">
            </td>
          </tr>
        </form>
      </table>
    </td>
  </tr>

```




```

        </tr>
</table>
</body>
</html>

```

Modul logout.php

```

<?
session_start();
session_destroy();
?>
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
</head>

<body>
<table width="100%" height="100%">
  <tr>
    <td align="center" valign="middle">
      You had been logged out from the system
    </td>
  </tr>
</table>
</body>
</html>

```

Modul xml_evy.php

```

<?php
// koneksi php ke mysql
include("connection/connect.php");

// $doc men-create obyek dom
$doc = new_xmldoc("1.0");

//header

header("Content-Type: text/xml");
header("Pragma: no-cache");

// $root untuk menyimpan variabel $doc yang digunakan untuk add
root (node node)
$root = $doc->add_root("ALLData");

//query database per-record
$query_cv = "SELECT ID_virus, nama_virus, keterangan_virus FROM
virus_komputer order by ID_virus asc";
$result_cv = mysql_query($query_cv) or die ("Error in query:
$query_cv. " . mysql_error());
while($row1 = mysql_fetch_object($result_cv))
{
  $record_cv = $root->new_child("virus_komputer", "");
}

```

```

//id_virus digunakan sebagai primary key
$record_cv->set_attribute("ID_virus", $row1->ID_virus);
$record_cv->new_child("nama_virus", $row1->nama_virus);
$record_cv->new_child("keterangan_virus", $row1->
>keterangan_virus);

$ID_virus_parent = $row1->ID_virus;

$query_srg = "SELECT bulan_aktif_virus, tempat_serang FROM serang,
sasaran_serang where (serang.ID_serang = sasaran_serang.ID_serang)
AND (serang.ID_virus = $ID_virus_parent)";
$result_srg = mysql_query($query_srg) or die ("Error in query:
$query_srg. " . mysql_error());
$record_srg = $record_cv->new_child("serang", "");
while($row2 = mysql_fetch_object($result_srg))
{
    $record_def = $record_srg->new_child("detail_serang", "");
    $record_def->new_child("bulan_aktif_virus", $row2->
>bulan_aktif_virus);
    $record_def->new_child("tempat_serang", $row2->tempat_serang);
}

$query_sol = "SELECT solusi FROM solusi,memiliki_solusi where
(solusi.ID_solusi = memiliki_solusi.ID_solusi) AND
(memiliki_solusi.ID_virus = $ID_virus_parent)";
$result_sol = mysql_query($query_sol) or die ("Error in query:
$query_sol. " . mysql_error());
$record_sol = $record_cv->new_child("solusi_virus", "");
while($row3 = mysql_fetch_object($result_sol))
{
    $record_sol->new_child("keterangan_solusi", $row3->solusi);
}

}

$strXML = $doc->dumppmem();
$dtd = "
<!DOCTYPE ALLData [
    <!ELEMENT ALLData (virus_komputer)+>
    <!ELEMENT virus_komputer (nama_virus, keterangan_virus,
serang, solusi_virus)>
    <!ELEMENT nama_virus (#PCDATA)>
    <!ELEMENT keterangan_virus (#PCDATA)>
    <!ELEMENT serang (detail_serang)*>
    <!ELEMENT detail_serang (bulan_aktif_virus, tempat_serang)>
    <!ELEMENT bulan_aktif_virus (#PCDATA)>
    <!ELEMENT tempat_serang (#PCDATA)>
    <!ELEMENT solusi_virus (keterangan_solusi)*>
    <!ELEMENT keterangan_solusi (#PCDATA)>
    <!ATTLIST virus_komputer ID_virus CDATA '0'>
]>";
$search = '<?xml version="1.0"?>';
$replace = $search . $dtd;

```

```
echo str_replace($search, $replace, $strXML);
?>
```

Halaman untuk *Client*

Modul Load.frm

```
'jk klik ok maka akan meng-load Menu_Utama
Private Sub Image2_Click()
    Unload Me
    Menu_Utama.Show
End Sub

'jk klik cancel maka sistem akan keluar
Private Sub Image3_Click()
    Unload Me
End Sub
```

Modul Setup.frm

```
'jk tombol ok ditekan
Private Sub CmdOk_Click()
    If TxtUrl.Text = "" Then
        'url/document xml path kosong
        MsgBox "XML URL/Document must not empty!", vbCritical,
            "Setup Error"
    Else
        'url set 1 (gagal)
        url = Trim(TxtUrl.Text)
        If frmVirus.ParseXMLDoc(TxtUrl.Text) = 1 Then
            'MsgBox "Load URL/Document failed!", vbCritical,
            '"Error Load file"
        Else
            'selain kejadian di atas berarti 0 (sukses)
            If URLType = "" Then
                URLType = "Online"
            End If

            frmVirus.Header1.Caption = "The Information of
            Computer Virus (" & URLType & ")"
            frmVirus.Load = True           'formVirus diload
            frmSetup.Visible = False      'formSetup dist false
            frmSearch.Visible = False     'formSearch diset false
            Menu_Utama.IOCV.Enabled = True 'Menu_Utama diset
            true/diaktifkan
        End If
    End If
End Sub

'cmd untuk ambil file xml secara lokal
Private Sub cmdXMLFile_Click()
    Menu_Utama.cdIlgMain.Flags = cdIOFNFileMustExist Or _
        cdIOFNHideReadOnly Or _
        cdIOFNPathMustExist
```

```

'menfilter semua file yang berekstensi xml
Menu_Utama.cdlgMain.Filter = "XML Files (*.xml)|*.xml"
'buka jendela open
Menu_Utama.cdlgMain.ShowOpen

If Menu_Utama.cdlgMain.FileName <> "" Then
    TxtUrl.Text = Menu_Utama.cdlgMain.FileName
    URLType = "local"
End If
End Sub

'jka tombol cancel ditekan, semua form setup dan Menu_Utama di non
aktifkan
Private Sub CmdCancel_Click()
    frmSetup.Visible = False
    Menu_Utama.IOCV.Enabled = False
End Sub

'untuk minimize maka di hide
Private Sub Form_KeyPress(KeyAscii As Integer)
    If KeyAscii = 27 Then
        Me.Hide
    End If
End Sub

'tombol alih
Private Sub TxtUrl_KeyDown(KeyCode As Integer, Shift As Integer)
    Call CekTombol(KeyCode, Me)
End Sub

'form load / cover diset center
Private Sub Form_Load()
    frmSetup.Left = Menu_Utama.Width / 2 - frmSetup.Width / 2
    frmSetup.Top = Menu_Utama.Height / 2 - frmSetup.Height / 2
End Sub

'jka tombol close ditekan maka form setup akan disembunyikan
Private Sub btnClose_Click()
    Me.Hide
End Sub

```

Modul Menu_Utama.frm

```

'fungsi MDIForm
Private Sub Menu_Utama_Load()
    On Error Resume Next
    IOCV.Enabled = False
End Sub

'ketika setup dipilih form setup diload dan ditampilkan
Private Sub Setup_Click()
    Load frmSetup
    frmSetup.Show
    IOCV.Enabled = False
End Sub

```

```

'jk IOCV di select, form virus ditampilkan
Private Sub IOCV_Click()
    frmVirus.Show
End Sub

Private Sub Exit_Click()
    Unload Me
End Sub

'load form about programmer di set center
Private Sub AboutProgrammer_Click()
    Load frmAboutProgrammer
    frmAboutProgrammer.Left = Menu_Utama.Width / 2 -
    frmAboutProgrammer.Width / 2
    frmAboutProgrammer.Top = Menu_Utama.Height / 2 -
    frmAboutProgrammer.Height / 2
    frmAboutProgrammer.Show
End Sub

'load form about program di set center
Private Sub AboutProgram_Click()
    frmAboutProgram.Left = Menu_Utama.Width / 2 -
    frmAboutProgram.Width / 2
    frmAboutProgram.Top = Menu_Utama.Height / 2 -
    frmAboutProgram.Height / 2
    frmAboutProgram.Show
End Sub

```

Modul Virus.frm

```

Option Explicit
Public ParseValue           As Integer
Public ObjDOMDocument      As MSXML.DOMDocument
Public ObjXMLDOMNode       As MSComctlLib.Node
Public ObjXMLDOMNodeList   As IXMLDOMNodeList
Public Load                 As Boolean
Public NextKey              As Long
Public W, i, J, X, Y1, Y2, Y3, Sign    As Integer
Dim Result                  As Long

' fungsi ini digunakan untuk mem-parse XML (cek untuk sukses dan
gagal)
' fungsi returns-nya:
' 0 - sukses
' 1 - gagal

Public Function ParseXMLDoc(Path As String) As Long
    Dim FailValue           As Long

    Set ObjDOMDocument = New MSXML.DOMDocument
    tvwXMLText.Nodes.Clear

    ObjDOMDocument.async = False 'berarti diset synchronous
    ObjDOMDocument.validateOnParse = True
    ObjDOMDocument.Load Path

```

```

If ObjDOMDocument.parseError.errorCode <> 0 Then ' check for
error Loading
    FailValue = 1
Else
    Set ObjXMLDOMNode = tvwXMLText.Nodes.Add(, , "ROOT",
"XML Document Contents")
    Set ObjXMLDOMNodeList =
ObjDOMDocument.selectNodes("//virus_komputer")
WalkChildNodes ObjDOMDocument.childNodes,
ObjXMLDOMNode

    'set J dan W untuk isi text ke grid
    J = 1
    W = 1
    'panggil fungsi untuk menampilkan data dalam grid
    WalkChildNodesToGrid ObjDOMDocument.childNodes

End If
End Function

Public Function WalkChildNodes(NodeList As MSXML.IXMLDOMNodeList,
_TreeViewParentNode As MSCComctlLib.Node) As Long

    Dim ObjElement          As MSXML.IXMLDOMNode
    Dim ObjAttribute        As MSXML.IXMLDOMAttribute
    Dim NewNode             As MSCComctlLib.Node

    For Each ObjElement In NodeList
        If Not (ObjElement.nodeName = "#text") Then
            NextKey = NextKey + 1

            If NextKey = 2 Then
                'do nothing
            Else

                Set NewNode =
                tvwXMLText.Nodes.Add(TreeViewParentNode.Key, tvwChild,
                _"key-" & CStr(NextKey))

                'jka childnotes nya > 1
                If (ObjElement.childNodes.length > 1) Then
                    'jka "virus_komputer maka akan diset nilai
                    'attribute nya
                    If ObjElement.nodeName = "virus_komputer" Then
                        'jka attribute tidak kosong
                        If Not (ObjElement.Attributes Is Nothing) Then
                            'NewNode.Text untuk set ID virus dengan
                            'isi nilai attribute-nya
                            For Each ObjAttribute In
                                ObjElement.Attributes
                                    NewNode.Text = ObjElement.nodeName & "
                                    (ID virus: " & ObjAttribute.Value &
                                    ")"
                                Next
                            End If
                        End If
                    End If
                End If
            End If
        End If
    Next
End Function

```

```

Else
    'selain di atas maka NewNode.Text diset untuk
    'menampilkan isi text saja pada treeview
    NewNode.Text = ObjElement.nodeName
End If
Else
    'jk ObjElement serang da solusi_virus maka akan
    'ditampilkan isi text dan nama tag nya
    If ObjElement.nodeName = "serang" Or
    ObjElement.nodeName = "solusi_virus" Then
        'Node name dan isinya ditampilkan
        NewNode.Text = ObjElement.nodeName
    Else
        'selain kondisi di atas maka isi text saja
        'yang ditampilkan
        NewNode.Text = ObjElement.Text
    End If
End If
'ini bisa pakai .tag atau .text, .tag biasanya
'digunakan sebagai cadangan selain .text
NewNode.Tag = ObjElement.Text

End If
'jk childnotes nya >= 1 untuk seleksi node di dalam
'node... jadi untuk node yang ada di dalam
If ObjElement.childNodes.length >= 1 Then

    WalkChildNodes ObjElement.childNodes, NewNode
End If
End If
Next
End Function

'fungsi treeview ke dalam text
Public Function WalkChildNodesToText(NodeList As
MSXML.IXMLDOMNodeList) As Long
    Dim ObjElement          As MSXML.IXMLDOMNode
    Dim ObjAttribute        As MSXML.IXMLDOMAttribute

    For Each ObjElement In NodeList
        If Not (ObjElement.nodeName = "#text") Then
            NextKey = NextKey + 1

            Select Case UCase(ObjElement.nodeName)
                Case "NAMA_VIRUS"
                    txtname.Text = ObjElement.Text
                Case "TEMPAT_SERANG"
                    txtTarget.Text = txtTarget.Text &
                    ObjElement.Text & ", "
                    'bulan aktif dapat banyak/many
                Case "BULAN_AKTIF_VIRUS"
                    cboactive.AddItem ObjElement.Text, i
                    i = i + 1
                    'isi nilai textnya dalam keadaan bayak/many
                    cboactive.Text = ObjElement.Text
                Case "KETERANGAN_VIRUS"

```

```

        txtExplain.Text = ObjElement.Text
    Case "SOLUSI_VIRUS"
        txtSolution.Text = ObjElement.Text
    End Select

    'jka attribute tidak kosong
    If Not (ObjElement.Attributes Is Nothing) Then
        For Each ObjAttribute In ObjElement.Attributes
            NextKey = NextKey + 1
        Next
    End If

    If ObjElement.childNodes.length > 0 Then
        WalkChildNodesToText ObjElement.childNodes
    End If
End If

If ObjElement.nodeName = "virus_komputer" Then
    #text elements are the text values in an element and can be
    extracted using the text value of an element
    Exit Function
End If

Next
End Function

'fungsi treeviewXML ke dalam grid
Public Function WalkChildNodesToGrid(NodeList As
MSXML.IXMLDOMNodeList) As Long
    Dim ObjElement As MSXML.IXMLDOMNode
    Dim ObjAttribute As MSXML.IXMLDOMAttribute
    For Each ObjElement In NodeList
        If Not (ObjElement.nodeName = "#text") Then
            NextKey = NextKey + 1

            'variabel - variabel yang ada digunakan untuk
            'meletakkan
            'data agar dapat diletakkan di grid
            Select Case UCase(ObjElement.nodeName)
                Case "NAMA_VIRUS"
                    'variabel untuk baris, selalu ditambah 1 untuk
                    'increment
                    W = W + 1
                    tvwXMLGrid.TextMatrix(J, 1) = ObjElement.Text
                    X = J
                    Y1 = X
                    Y2 = X
                    Y3 = X
                Case "TEMPAT_SERANG"
                    J = J + 1
                    Y1 = Y1 + 1
                    If Y1 > tvwXMLGrid.Rows Then
                        Y1 = Y1 - 1
                        J = J - 1
                    End If
                    tvwXMLGrid.Rows = J + 1
                    tvwXMLGrid.TextMatrix(Y1 - 1, 2) =
                    ObjElement.Text
            End Select
        End If
    Next
End Function

```



```

        J = Y1
        Case "BULAN_AKTIF_VIRUS"
            J = J + 1
            Y2 = Y2 + 1
            If Y2 > tvwXMLGrid.Rows Then
                Y2 = Y2 - 1
                J = J - 1
            End If
            tvwXMLGrid.Rows = J + 1
            tvwXMLGrid.TextMatrix(Y2 - 1, 3) =
            ObjElement.Text
            If Y2 > J Then J = Y2
        Case "KETERANGAN_VIRUS"
            tvwXMLGrid.TextMatrix(X, 4) = ObjElement.Text
        Case "KETERANGAN"
            J = J + 1
            Y3 = Y3 + 1
            If Y3 > tvwXMLGrid.Rows Then
                Y3 = Y3 - 1
                J = J - 1
            End If
            tvwXMLGrid.Rows = J + 1
            tvwXMLGrid.TextMatrix(Y3 - 1, 5) =
            ObjElement.Text
            If Y3 > J Then J = Y3
    End Select

    If Not (ObjElement.Attributes Is Nothing) Then
        For Each ObjAttribute In ObjElement.Attributes
            NextKey = NextKey + 1
        Next
    End If

    If ObjElement.childNodes.length > 0 Then
        WalkChildNodesToGrid ObjElement.childNodes
    End If
End If
Next
End Function

' fungsi untuk membuat grid
Private Sub Form_Load()
    'center
    frmVirus.Left = Menu_Utama.Width / 2 - frmVirus.Width / 2
    frmVirus.Top = Menu_Utama.Height / 2 - frmVirus.Height / 2

    'lebar kolom
    tvwXMLGrid.ColWidth(0) = 400
    tvwXMLGrid.ColWidth(1) = 1000
    tvwXMLGrid.ColWidth(2) = 3000
    tvwXMLGrid.ColWidth(3) = 3000
    tvwXMLGrid.ColWidth(4) = 3000
    tvwXMLGrid.ColWidth(5) = 3000

    'isi nama kolom
    tvwXMLGrid.TextMatrix(0, 1) = "VIRUS NAME"

```

```

tvwXMLGrid.TextMatrix(0, 2) = "TARGET"
tvwXMLGrid.TextMatrix(0, 3) = "ACTIVE TIME"
tvwXMLGrid.TextMatrix(0, 4) = "EXPLAIN"
tvwXMLGrid.TextMatrix(0, 5) = "SOLUTION"
End Sub

'fungsi dari tvwXMLGrid ke dalam grid
Private Sub tvwXMLGrid_Click()
    Dim ObjSelGrid As String
    'mengambil propertis - propertis di node yang dipilih
    ObjSelGrid = tvwXMLGrid.TextMatrix(tvwXMLGrid.Row, 1)
    txtTarget.Text = ""
    txtname.Text = ""
    txtExplain.Text = ""
    txtSolution.Text = ""
    'membersihkan isi msflexgrid
    cboactive.Clear
    i = 0
End Sub

'fungsi tvwXMLText ke dalam textbox
Private Sub tvwXMLText_Click()
    Dim ObjSelNode As Node
    'mengambil propertis - propertis di node yang dipilih
    Set ObjSelNode = tvwXMLText.SelectedItem
    txtTarget.Text = ""
    txtname.Text = ""
    txtExplain.Text = ""
    txtSolution.Text = ""
    'membersihkan isi textbox
    cboactive.Clear
    i = 0
    'memanggil fungsi yang akan digunakan untuk mengisi teks box
    'dengan data node yang terpilih
    'panggil fungsi PopulateVirusDetails
    PopulateVirusDetails ObjSelNode
End Sub

' fungsi ini digunakan untuk nge-list blok warna text
' detail informasi virus di dalam textbox
Public Sub PopulateVirusDetails(ObjSelNode As Node)
    Dim ObjXMLDOMNodeList As IXMLDOMNodeList
    Dim ObjChildElement As MSXML.IXMLDOMNode
    Dim Text1 As String

    'jk ObjSelNodenya kosong maka secara otomatis diakhiri
    If ObjSelNode Is Nothing Then Exit Sub

    If Trim(ObjSelNode.Tag) <> "" Then
        If tvwXMLText.SelectedItem.Child Is Nothing Then
            Else
                Text1 = tvwXMLText.SelectedItem.Child.Text
            End If
        End If
    End If

```

```

Set ObjXMLDOMNodeList =
ObjDOMDocument.selectNodes("//ALLData//virus_komputer[nama_v
irus= '" & Text1 & "']")
' panggil fungsi untuk ke text
WalkChildNodesToText ObjXMLDOMNodeList
End If
End Sub

'fungsi pada cbosearch
Private Sub cbosearch_Click()
'berdasarkan virus_name
If (cbosearch.ListIndex = 0) Then
frmSearch.Label10.Caption = "Enter The Name Of Computer
Virus : "
frmSearch.Show
ParseValue = cbosearch.ListIndex
'berdasarkan active time
ElseIf (cbosearch.ListIndex = 1) Then
frmSearch.Label10.Caption = "Enter The Month Of Aktive Time
: "
frmSearch.Show
ParseValue = cbosearch.ListIndex
'berdasarkan target
ElseIf (cbosearch.ListIndex = 2) Then
frmSearch.Label10.Caption = "Enter The Target Of Computer
Virus : "
frmSearch.Show
ParseValue = cbosearch.ListIndex
Else
' another conditions
End If
frmSearch.txtsearch.Text = ""
End Sub

'fungsi untuk cari berdasarkan pilihan yang dikehendaki,
'setelah memilih sesuai yang dikehendaki maka fungsi ini akan
'memanggil
'fungsi berdasar pilihan yang dikehendaki
Public Sub SearchActions(Index As Integer)
If Load = False Then
MsgBox "XML Doc not yet Loaded" 'jka false maka msgbox
'ini dijalankan
Else
Select Case Index
Case 0
tvwXMLText.Nodes.Clear
'pertama, tambah "root" node to treeview
'ini set untuk treeview
Set ObjXMLDOMNode = tvwXMLText.Nodes.Add(, ,
"ROOT", "XML Document Contents")
'skrng men-retrieve node list menggunakan XPath
'query
'karena telah diset untuk treeview maka set untuk
'xml yang akan diletakkan dalam
'ObjXMLDOMNode yang telah diset

```

```

        Set ObjXMLDOMNodeList =
        ObjDOMDocument.selectNodes("//virus_komputer{nama_virus= '" & Trim(frmSearch.txtsearch.Text) &
        "'}")
        'panggil fungsi WalkChildNodes
        WalkChildNodes ObjXMLDOMNodeList, ObjXMLDOMNode

    Case 1
        tvwXMLText.Nodes.Clear
        'pertama, tambah "root" node to treeview
        Set ObjXMLDOMNode = tvwXMLText.Nodes.Add(, ,
        "ROOT", "XML Document Contents")
        Set ObjXMLDOMNodeList =
        ObjDOMDocument.selectNodes("//virus_komputer")
        'panggil fungsi WalkChildNodes
        WalkChildNodesForSearch ObjXMLDOMNodeList,
        ObjXMLDOMNode

    Case 2
        tvwXMLText.Nodes.Clear
        'pertama, tambah "root" node to treeview
        Set ObjXMLDOMNode = tvwXMLText.Nodes.Add(, ,
        "ROOT", "XML Document Contents")
        Set ObjXMLDOMNodeList =
        ObjDOMDocument.selectNodes("//virus_komputer")
        'panggil fungsi WalkChildNodes
        WalkChildNodesForSearch ObjXMLDOMNodeList,
        ObjXMLDOMNode

    End Select
End If

End Sub

'fungsi untuk cari berdasarkan active time dan target
'fungsi ini logikanya, jika ditemukan apa yang dicari maka node
'yang lainnya akan dihapus/remove
'dan jika tidak ditemukan maka semua node akan dihapus

Public Function WalkChildNodesForSearch(NodeList As
MSXML.IXMLDOMNodeList, _TreeViewParentNode As MSComctlLib.Node) As
Long

    Dim ObjElement          As MSXML.IXMLDOMNode
    Dim ObjAttribute        As MSXML.IXMLDOMAttribute
    Dim NewNode             As MSComctlLib.Node

    For Each ObjElement In NodeList
        If Not (ObjElement.nodeName = "#text") Then
            NextKey = NextKey + 1
            Set NewNode =
            tvwXMLText.Nodes.Add(TreeViewParentNode.Key, tvwChild,
            _"key-" & CStr(NextKey))

            If ObjElement.childNodes.length >= 1 Then
                WalkChildNodesForSearch ObjElement.childNodes, NewNode
            End If
        End If
    Next ObjElement
End Function

```

```

End If

If (ObjElement.childNodes.length > 1) Then
  If ObjElement.nodeName = "virus_komputer" Then
    If Not (ObjElement.Attributes Is Nothing) Then
      For Each ObjAttribute In
        ObjElement.Attributes
          'ini set 1 jk berhasil ditemukan apa
          'yang dicari
          If Sign = 1 Then
            NewNode.Text = ObjElement.nodeName &
              " (ID virus: " & ObjAttribute.Value
              & ")"
            'set o lagi biar reset dari awal,
            'cari dari awal lagi
            Sign = 0
          Else
            tvwXMLText.Nodes.Remove
              NewNode.Index
          End If
        Next
      End If
    Else
      If Sign = 1 Then
        NewNode.Text = ObjElement.nodeName
      End If
    End If
  Else
    If ObjElement.nodeName = "serang" Or
      ObjElement.nodeName = "solusi_virus" Then
      NewNode.Text = ObjElement.nodeName
    Else
      Result = InStr(1, ObjElement.Text,
        Trim(frmSearch.txtsearch.Text), vbBinaryCompare)
      If Result <> 0 Then
        Sign = 1
      End If
      NewNode.Text = ObjElement.Text
    End If
  End If

  NewNode.Tag = ObjElement.Text
End If
Next
End Function

' fungsi untuk show all treview
Private Sub cmdShowAll_Click()
  'membersihkan isi textbox
  tvwXMLText.Nodes.Clear
  'manggil fungsi parseXMLdoc url
  NextKey = 0
  ParseXMLDoc (url)
End Sub

```

```
' pada saat di close berarti hide bukan exit
Private Sub Close_Click()
    Me.Hide
End Sub
```

Modul Search.frm

```
'jka tombol search ditekan, frmSearch disembunyikan
'manggil fungsi untuk SearchActions
Private Sub SearchButton_Click()
    frmSearch.Hide
    Call frmVirus.SearchActions(frmVirus.ParseValue)
End Sub
```

```
'fungsi buat center untuk load
Private Sub Form_Load()
    frmSearch.Left = Menu_Utama.Width / 2 - frmSearch.Width / 2
    frmSearch.Top = Menu_Utama.Height / 2 - frmSearch.Height / 2
End Sub
```

Modul AboutProgram.frm

```
'jka exit maka di hide
Private Sub ExitButton1_Click()
    Me.Hide
End Sub
```

Modul AboutProgrammer.frm

```
Private Sub Form_Load()
    Image1.Picture = Menu_Utama.ImageList1.ListImages(6).Picture
End Sub
```

```
'jka di exit maka di hide
Private Sub ExitButton2_Click()
    Me.Hide
End Sub
```

